AUTHORIZED DOCUMENTATION

Administration Guide

Novell® SecureLogin

6.1 SP1

June, 2009

www.novell.com

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the Novell International Trade Services Web page (http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed on the Novell Legal Patents Web page (http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the latest online documentation for this and other Novell products, see the Novell Documentation Web page (http://www.novell.com/documentation).

Novell Trademarks

For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide

This guide provides you information to configure and manage Novell® SecureLogin users in multi-user and standalone environments.

This document consists of the following sections:

Audience

This guide is intended for:

- Network Administrators
- Systems Administrators
- IT Support Staff

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to the Novell Documentation Feedback (http://www.novell.com/documentation/feedback.html) and enter your comments there.

Documentation Updates

For the most recent version of the *Novell SecureLogin 6.1 SP1 Administration Guide*, visit the Novell Documentation Web site. (http://www.novell.com/documentation/securelogin61/index.html)

Additional Documentation

The *Administration Guide* is part of the documentation set for Novell® SecureLogin 6.1.

Other documents include:

- *Novell SecureLogin 6.1 SP1 Installation Guide*
- *Novell SecureLogin 6.1 SP1 Application Definition Guide*
- *Novell SecureLogin 6.1 SP1 Citrix and Terminal Services Guide*
- *Novell SecureLogin 6.1 SP1 User Guide*
- Quick Start. *NMAS Login Method and Login ID Snap-In for pcProx*
- Readme. Available online at *Novell SecureLogin 6.1 SP1 Administration Guide*, visit the Novell Documentation Web site. (http://www.novell.com/documentation/securelogin61sp1/index.html)

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

Getting Started

1

Before you begin working on Novell SecureLogin, you should have a strong working knowledge of the following:

- Microsoft* Active Directory*
- Microsoft Management Console (MMC)
- Microsoft Group Policy Object Management Console (GPMC)
- Microsoft Windows operating systems
- Lightweight Directory Access Protocol (LDAP)

Novell SecureLogin consists of the Personal Management utility, Personal Management utility, and plug-in for inclusion in Novell SecureLogin, which are used for administering Novell SecureLogin

The Personal Management Utility

The end-users of the Novell SecureLogin use the Personal Management utility to customize Novell SecureLogin to their requirements and preferences.

The Administration Management Utilities

The administrators use the Novell SecureLogin to manage and administer Novell SecureLogin.

Additional Plug-In

The SSO additional plug-in is available in iManager, one of the Administrative Management utilities.

The plug-in is used to manage the single sign-on features.

The following sections explain these utilities and the accessing the plug-in:

- Section 1.1, “Personal Management Utility,” on page 13
- Section 1.2, “Starting the Administrative Management Utilities,” on page 14
- Section 1.3, “Accessing the Single Sign-On Plug-In Through iManager,” on page 15

1.1 Personal Management Utility

The end-users of Novell SecureLogin use the Personal Management utility to configure and customize Novell SecureLogin to their preferences and requirements.

To start the Personal Management utility, do one of the following:

- Double-click the Novell SecureLogin  icon in the notification area.
- In the notification area, right-click the Novell SecureLogin  icon and select *Manage Logins*.
- On the Windows *Start* menu, select *Programs* > *SecureLogin* > *Novell SecureLogin.*

NOTE: Changes made by using the Personal Management utility on the local workstation apply only to the logged-in user and override settings made in the directory.

For example, if the *Allow users to view and modify Application Definitions* preference is set to *No*, at the organizational unit level, and set to *Yes* on the actual user object in the directory, then the user object setting applies and the user can view and modify application definitions. However, other users in the container cannot view and modify application definitions unless they have the option set on their user objects.

1.2 Starting the Administrative Management Utilities

The Novell SecureLogin built-in Administrative Management utility contains additional functionality that is not included in the Personal Management utility.

Use the Administrative Management utility for LDAP-compliant directories:

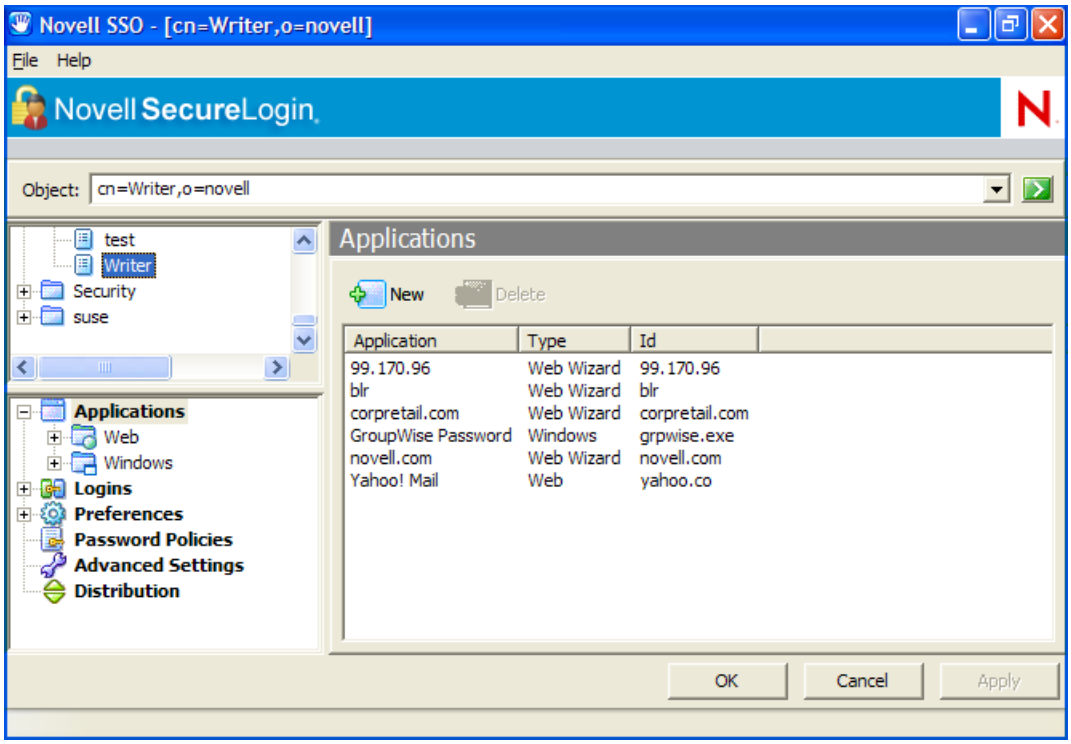
To access the Administrative Management utility:

1. On the *Start* menu, select *Programs* > *Novell SecureLogin* > *SecureLogin Manager*. The Administrative Management utility is displayed.

   or,

   Double-click `slmanager.exe` (by default, it is in the `C:\Program Files\Novell\SecureLogin\Tools` directory).

   The SecureLogin Manager page is displayed.
2. In the *Object* field, specify your object name, then press the Enter key.



You must press the Enter key to submit the entry typed in the *Object* field. Clicking *OK* closes the dialog box but does not accept the entry you typed. The object name should be in the LDAP convention (username, objectname) if you are using LDAP mode and in the eDirectory™ convention (username. objectname), if you are using the eDirectory mode.

1.3 Accessing the Single Sign-On Plug-In Through iManager

The plug-in facilitates the administration of the Administrative Management utility: iManager for the administrators.

The iManager plug-in for Novell SecureLogin are `.npm` files. The plug-in are:

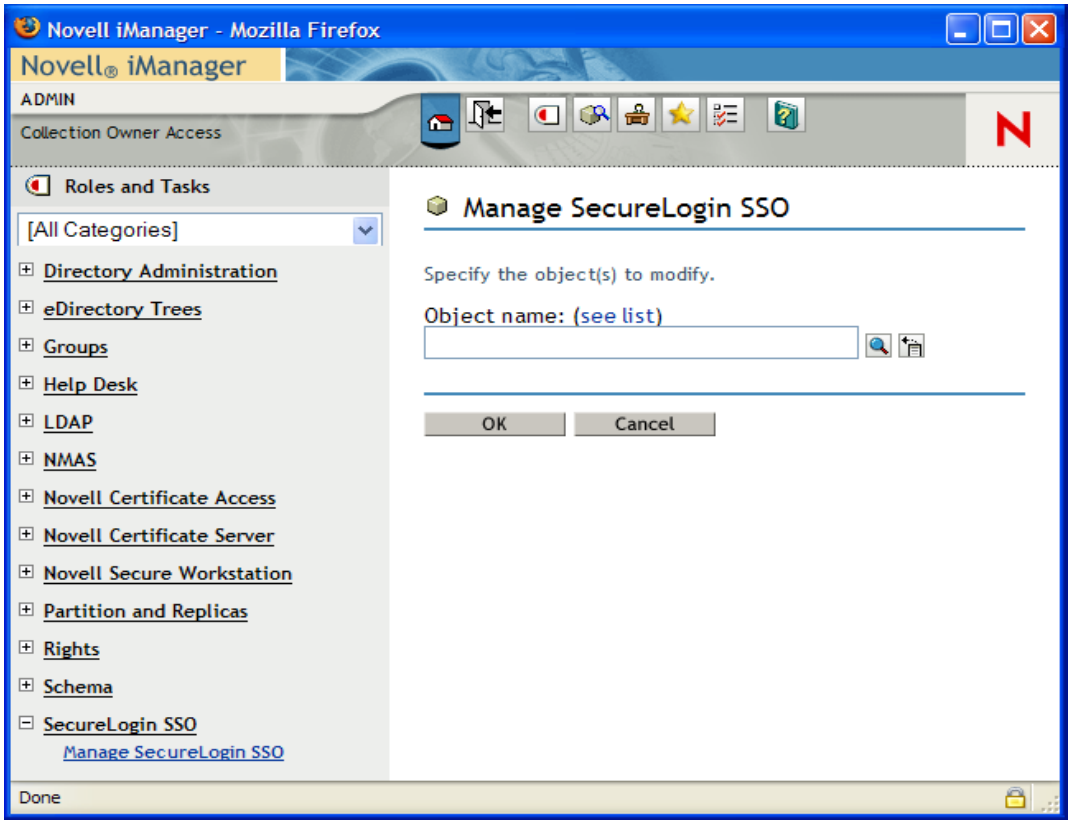
- `pcprox.npm`
- `secretstore.npm`
- `sso.npm`
- `sw.npm`

For more information on the npms, see "Installing the Other Plug-In for iManager " in the *Novell SecureLogin 6.1 SP1 Installation Guide.*

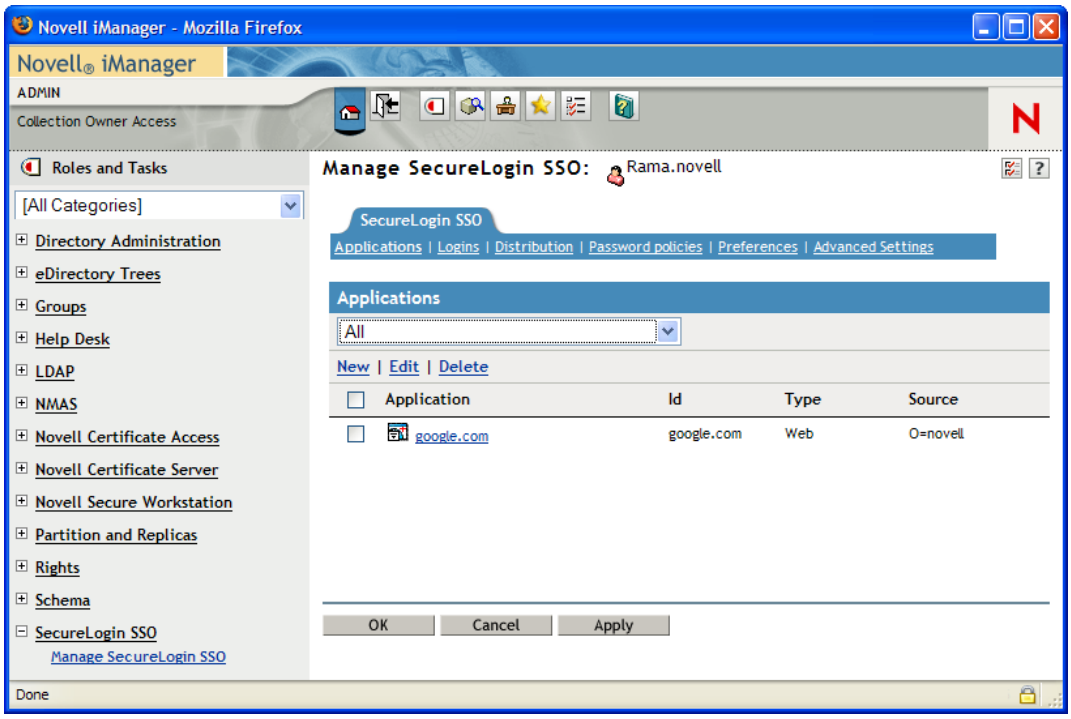
NOTE: Novell SecureLogin now supports iManager 2.6 and 2.7. The plug-in for iManager 2.6 are available as part of the product installer package. You can download the plug-in for iManager 2.7 from the Novell Downloads Web site. (http://download.novell.com/index.jsp)

To access the single sign-on plug-in through iManager:

1. Log in to iManager
2. Select *SecureLogin SSO* > *Manage SecureLogin SSO*. The Manage SecureLogin page is displayed.



3 In the *Object* field, specify your object name, then click *OK*. The Administrative Management page is displayed.



Configuring Novell SecureLogin

2

Novell SecureLogin 6.0 introduced a range of security features, including storing the single sign-on credentials on the user's smart card, encrypting the datastore by using the Public Key Infrastructure (PKI)-based credentials and the Advanced Encryption Standard (AES) encryption algorithm support.

To support the new features, you must change the Novell SecureLogin 6.0 datastore format.

The Novell SecureLogin 6.0 client can read data created by all the previous versions of Novell SecureLogin. However, the older versions cannot read the data created by version 6.0 and later. If the mixed corporate environment where some workstations are running Novell SecureLogin 6.0 or 6.1 and, other workstations are running previous versions, then data compatibility issues arise when a user moves between different versions of Novell SecureLogin on different workstations. This is particularly problematic in Citrix* environments or in large enterprise deployments.

If Novell SecureLogin 3.5 is present when you are installing Novell SecureLogin 6.1 SP1, it detects that version 3.5 data is in use and continues to function correctly. In this mode, version 3.5 functions are available. However, any new function that relies on version 6.1 data, is not available.

If you require the new functions, complete the following processes:

1. Choose a section of the tree to upgrade.

   For example,

   - Group
   - Container
   - Organization
   - User

2. Make sure that all user workstations in that section of the tree are upgraded with the Novell SecureLogin 6.1 SP1 client.

The next time the users log in, their data is converted to version 6.0 format and the new functions are available.

Complete the following tasks to configure Novell SecureLogin:

- Section 2.1, "Disabling User Access," on page 19
- Section 2.2, "Changing the Directory Datastore," on page 20
- Section 2.3, "Deleting or Re-setting User Data," on page 24

2.1 Disabling User Access

By default, the user has permission to change application definitions and predefined applications, passwords, and functionality.

You do this through the administrative management utilities.

This includes:

- Full access to all administrative tools.
- Access to selected administrative tools.
- Hiding the SecureLogin icon on the notification area.
- Hiding and password protecting the SecureLogin icon in the notification area.

If the SecureLogin icon is password protected, anyone attempting to access the Personal Management utility through the SecureLogin icon is prompted to provide the network password. This prevents non-authorized users from viewing SecureLogin data. However, the authorized user can use the administration tools to modify SecureLogin.

You can restrict access by setting preferences at the user, group policy, container, or organizational user (ou) level.

2.2 Changing the Directory Datastore

When the directory is upgraded, the new feature of Novell SecureLogin 6.0 are not available on the workstation. So, users must upgrade to the new version.

You can configure directory datastore version at the group policy, user object, container, or organizational unit levels.

We recommend that you set the datastore version at the container or the organization unit levels. This helps enterprises manage the datastore base and minimize the possibility of conflicting versions.

If you require to update a single new feature of Novell SecureLogin preference that requires 6.0 datastore, for example, when you upgrade the *Use AES for SSO data encryption* preference, you are prompted with a warning before proceeding to change.

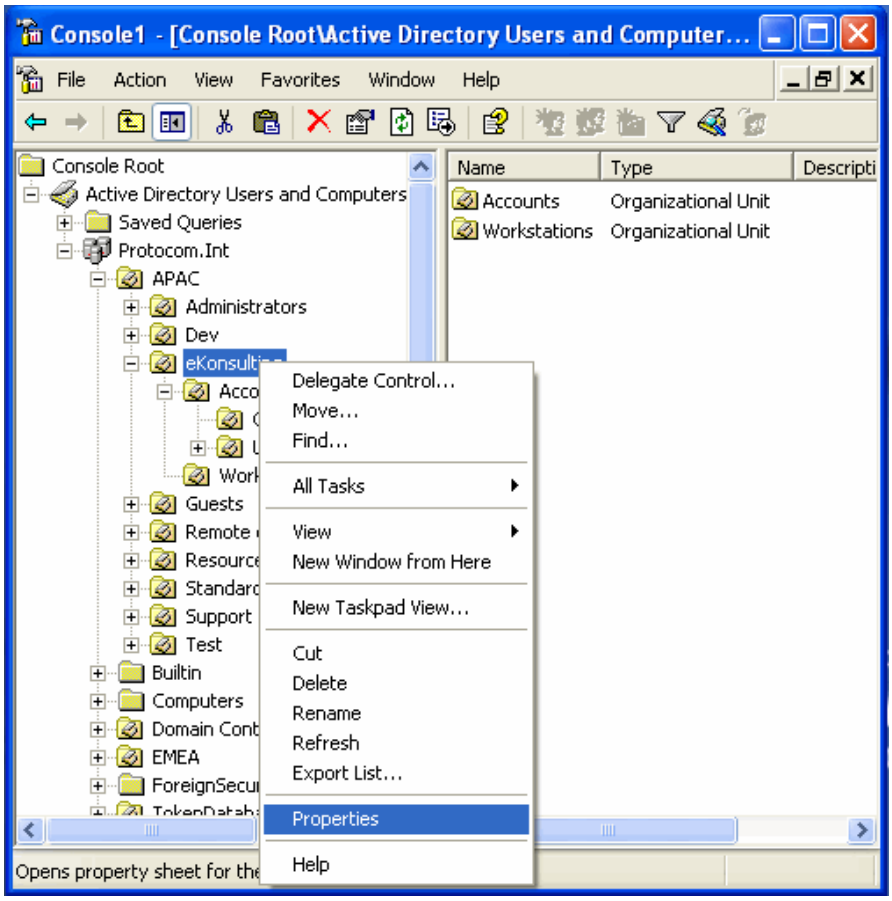
2.2.1 Changing the Organizational Unit Level Datastore

- "Changing the Organizational Unit Level Datastore in an Active Directory Environment" on page 20
- "Changing the Organizational Unit Level Datastore in an eDirectory Environment" on page 23
- "Deploying an Upgrade" on page 24

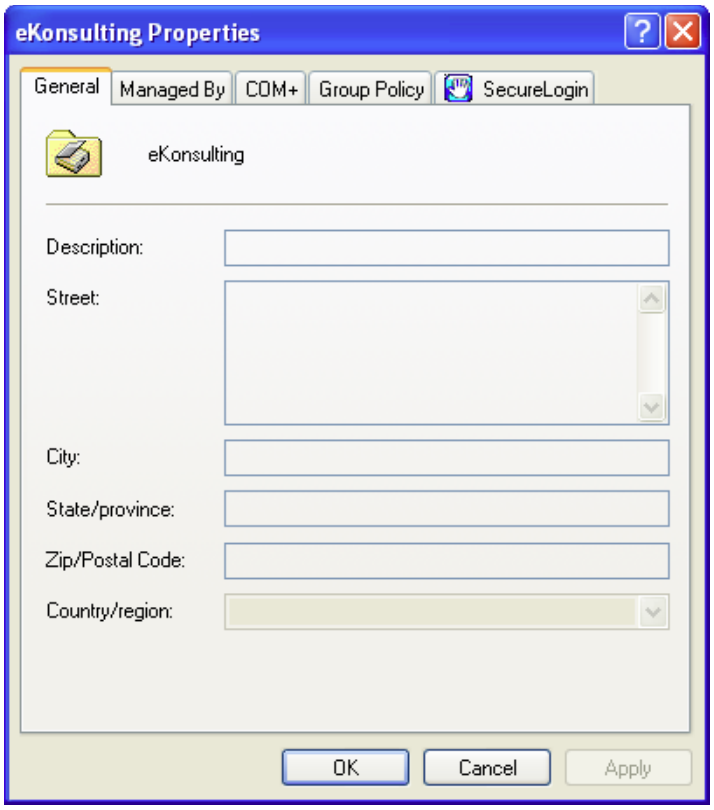
Changing the Organizational Unit Level Datastore in an Active Directory Environment

Do the following to set the directory datastore version at the organizational unit level in an Active Directory environment:

1. On the Windows *Start* menu, select *Programs* > *Administrative Tools* > *Active Directory Users and Computers*. The Microsoft Management Console is displayed.



2 Right click the required group policy, container, or OU, then click *Properties* (in this example, eKonsulting.) The properties dialog box is displayed.

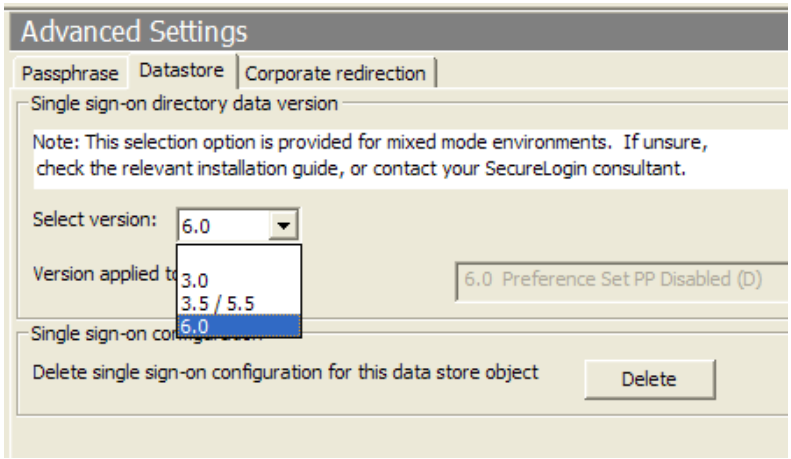


3 Click the *SecureLogin* tab. The SecureLogin page is displayed.

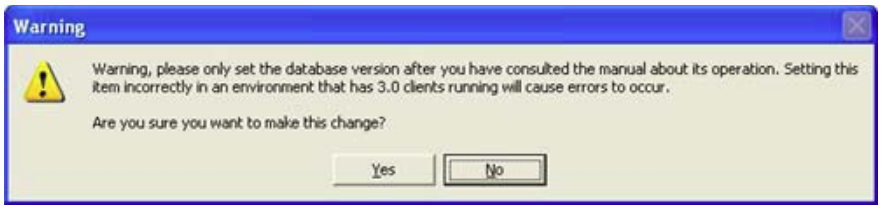
4 Click *Manage*. The Advanced Settings page of the administrative management (SecureLogin Manager) utility is displayed

5 On the left pane, click *Advanced Settings*. The Advanced Settings page is displayed.

6 Click the *Datastore* tab.



7 From the *Select version* drop-down list, select the required version. A warning is displayed.

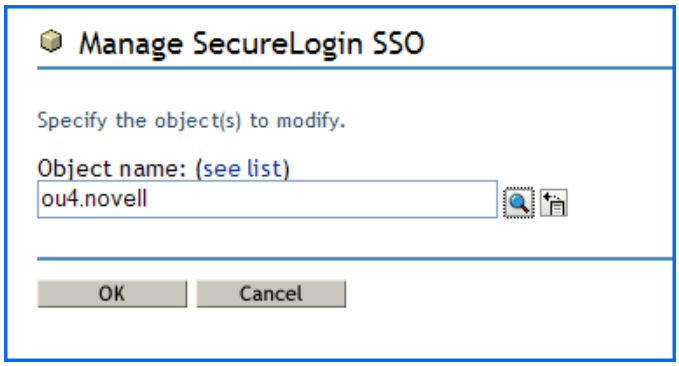


The warning message refers to 3.0 clients. This warning message is the same, and results in the same errors, if you are running version 3.5 or 5.5 clients for some users and then upgrade the datastore mode to version 6.0.

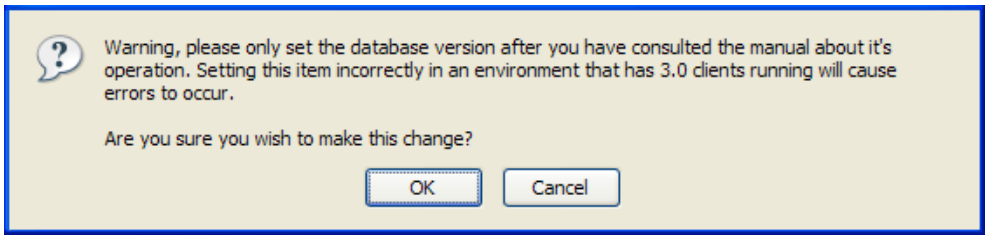
When a user's directory data version is upgraded, the datastore information displayed in the Novell SecureLogin About box is not updated until the user activates *Refresh Cache* from the *Advanced* menu of the Novell SecureLogin icon on the notification area, the next time he or she logs ins

Changing the Organizational Unit Level Datastore in an eDirectory Environment

1. Log in to iManager.
2. Specify the organizational unit object.
3. Click *OK*.



4. Click *Advanced Settings*. The Advanced Settings page is displayed.
5. From the *Select version* under the *Datastore* section, select the required version. A warning is displayed.



6. Click *OK* to make the changes.

Deploying an Upgrade

When you are deploying an upgrade across a series of workstations, follow the procedure explained in Section 2.2, “Changing the Directory Datastore,” on page 20. The next time the directory server and the workstation caches are synchronized and SecureLogin operates in the new version mode.

2.3 Deleting or Re-setting User Data

If a user has forgotten a network password and the passphrase answer or if the login credential data is corrupted, you must delete all SecureLogin data.

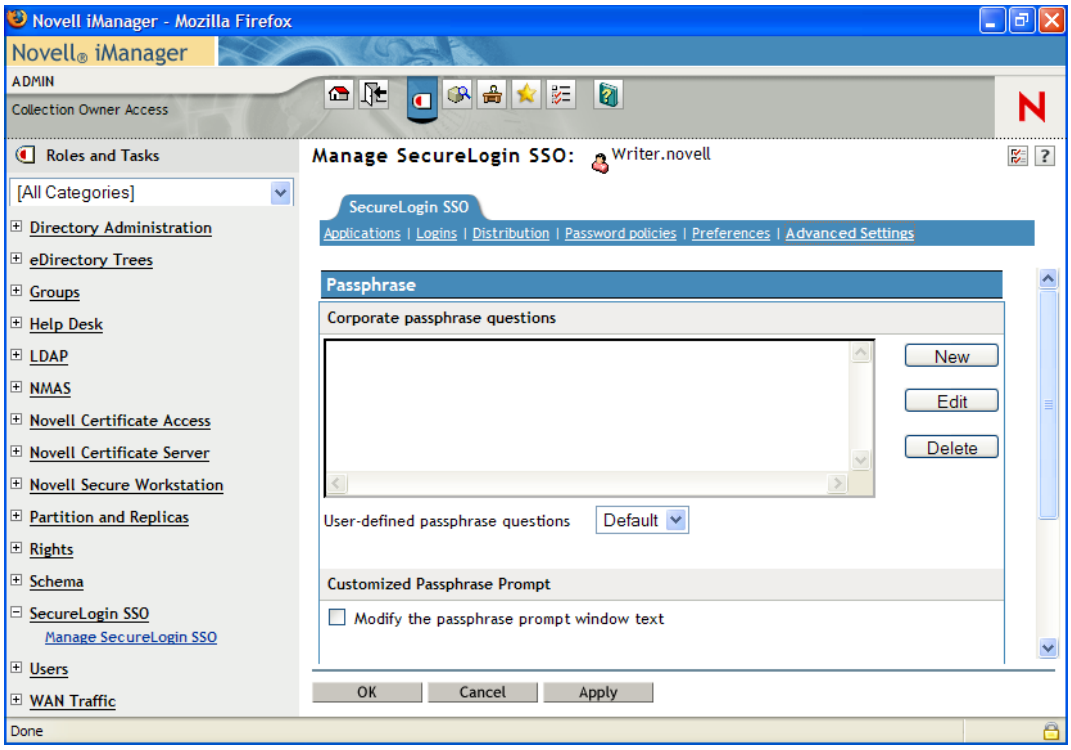
You as an administrator must do this because the user does not have access to the administrative management utilities.

To reset the user data:

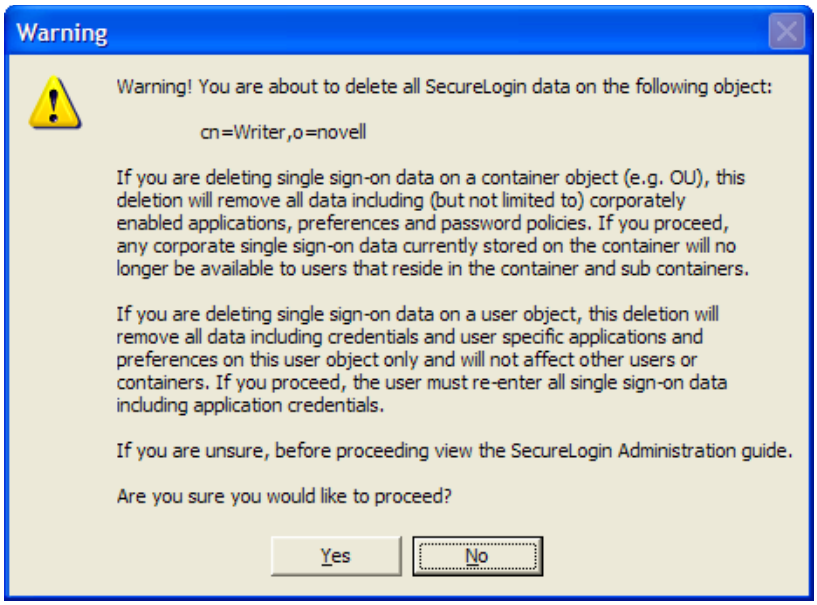
1. Access the Administrative Management utility.

   For information on accessing the Administrative Management utility see, Section 1.2, “Starting the Administrative Management Utilities,” on page 14 and, or, Section 1.3, “Accessing the Single Sign-On Plug-In Through iManager,” on page 15.

2. If you are using iManager, browse to *SecureLogin SSO* > *Manage SecureLogin SSO* > *Advanced Settings*. The Advanced Settings page is displayed.

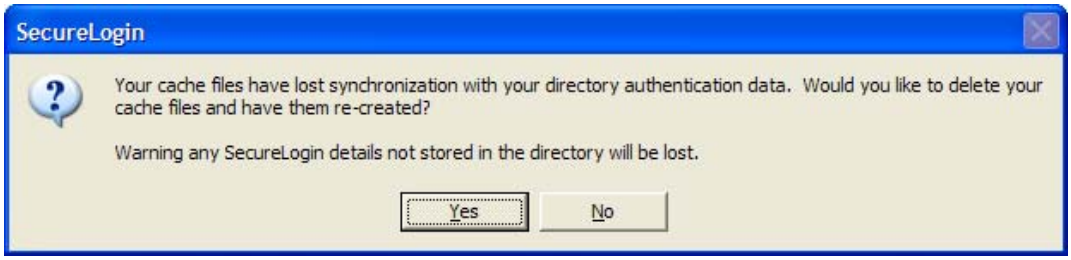


3. Click *Delete* in the Datastore section. A warning message appears.



4 Click *Yes*. The Datastore object is deleted.

If you did not delete the SecureLogin cache from the local cache, before you deleted the Datastore object data, you get an error message.



5 Click *Yes*.

NOTE: The next time the user logs on, the user will be asked to set up the passphrase question and response you configured and re-enter the credentials for each single sign-on-enabled application.

When you do this, you delete the complete data of the user, including:

- Credentials, including usernames and passwords
- Application definitions
- Predefined applications
- Password policies
- Preferences
- Passphrase questions and answers

WARNING: The deleted data is cannot be retrieved.

Before you delete a user's datastore object, consider the following important aspects:

User Data Re-set Option	Action
Select the required directory object only	The Delete single sign-on configuration for this datastore object option is available at the container, group policy, ou, and user object level.
Record (external to SecureLogin) all usernames, password, and additional required credential information	For example, if you delete a single sign-on-enabled application at the ou level, you might also be deleting the credentials for all users that reside in that container.
Delete the local cache on the workstation	The object or user continues to inherit configuration from higher-level objects in the directory even though you deleted the user data in the directory cache. This means that you should delete the local cache on the workstation first. This ensures that it does not synchronize with the directory cache and re-create the configuration in the directory.

The next time the user logs in, he or she is asked to set up the passphrase question and answer. They must re-enter the credentials for each single sign-on enabled application.

Managing Preferences

3

Novell SecureLogin preferences are tools, options, and parameters used by the enterprise administrators to configure the user's Novell SecureLogin corporate environment.

You can restrict a user's access to his or her Novell SecureLogin preferences through the administrative management utilities.

The preferences also include applications that are permitted to be enabled for single sign-on and the tools to enable users to access their own Novell SecureLogin management and administration functions.

You can configure user preferences from the Preference properties table in the administrative management utilities.

Prior to configuring Novell SecureLogin, we recommend that you refer Chapter 2, "Configuring Novell SecureLogin," on page 19 and ensure that you have completed the tasks explained in the section.

This section contains information on the following:

3.1 Preferences Categories

The Novell SecureLogin preferences are divided into the following categories:

- *General*
- *Java*
- *Security*
- *Web*
- *Windows*

3.2 The Default Preference Values

Each preference value has a default value that is implemented during installation or deployment. You can configure alternative values.

3.2.1 Inheriting Preference Values

In corporate directory hierarchies, preference values are inherited from higher-level objects, while some lower-level objects can override preferences set at higher-levels.

Therefore, the preference values set at the user object-level override all higher-level object values.

3.3 Setting User Preferences

You can set the SecureLogin user preferences in the Preferences Properties table in the Administrative Management utilities (Novell iManager or SecureLogin Manager).

Each SecureLogin preference has a default value that is implemented until an alternative value is manually configured. In directory hierarchies, preference values are inherited from a higher-level object, while some lower-level objects can override preference set at higher level.

For example, preference values set at the user object level override all higher level object values.

NOTE: This can be controlled for users by restricting their ability to set preferences.

For more information about inheriting configuration settings, see Chapter 13, "Distributing Configurations," on page 123.

3.4 Changing Preference Value

To change the preference value, do the following:

1. Access the Administrative Management utility.

   For information on accessing the Administrative Management utility, see Section 1.2, "Starting the Administrative Management Utilities," on page 14 and, or, Section 1.3, "Accessing the Single Sign-On Plug-In Through iManager," on page 15.
2. Click *Preference*. The Preference properties table is displayed.
3. Locate the setting you want to change and then, in the *Value* column, select the appropriate value.

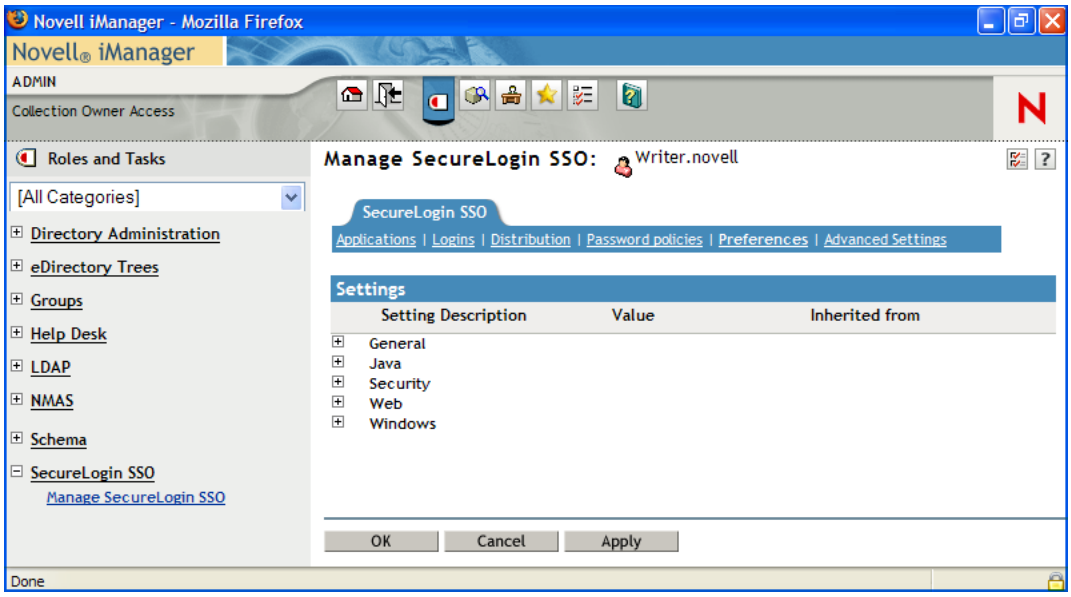
   NOTE: Some of the value settings are text field entries where you have to provide the value.

   ---
4. Click *OK*. The selected value is saved and the Administrative Management utility closes.

3.5 Setting the Preferences

You set preferences for managing SecureLogin in the Administration Management utility:

1. Log in to iManager.
2. Click *SecureLogin SSO* > *Manage SecureLogin SSO* > *Preferences*. The list of preferences is displayed.



3 Make the changes you want, then click *OK*.

Use the information in the following tables to assist you in making the changes:

- Table 3-1, "The General Preferences Properties Table," on page 30
- Table 3-2, "The Java Preferences Properties Table," on page 38
- Table 3-3, "The Security Preferences Properties Table," on page 38
- Table 3-4, "The Web Preferences Properties Table," on page 42
- Table 3-5, "The Windows Preferences Properties Table," on page 43

Changes in Preferences

This release of Novell SecureLogin has modified the *Allow users to view and modify application definitions preference*. This preference is now divided into two preferences:

- *Allow application definition to be modified by users*
- *Allow application definition to be viewed by users*

When you are upgrading from previous versions of Novell SecureLogin to version 6.1 by using a legacy directory data (6.0 or 3.5), if the *Allow users to view and modify application definitions option* was set to *No*, then the *Allow application definition to be modified by users* for 6.1 is dimmed.

You must reset the *Allow application definition to be viewed by users* to *Yes* before users can modify the application definitions.

Table 3-1 *The General Preferences Properties Table*

Preference	Value	Description
Allow "Close" option via system tray	*Yes*/*No*/*Default*	If the option is set to *No*, the *Close* option is not displayed and accessible in the Novell SecureLogin notification area icon. If this option is set to *Yes* or *Default*, the *Close* option is displayed and accessible in the Novell SecureLogin notification area icon. The default value is *Yes*. This preference is a available only through the administrative management utilities.
Allow "Log Off" option via system tray	*Yes*/*No*/*Default*	If this option is set to *No*, the *Log Off User* option is not displayed and accessible in the Novell SecureLogin notification area icon. If this option is set to *Yes* or *Default*, the *Log Off User* option is not displayed and accessible in the Novell SecureLogin notification area icon. The default value is *Yes*. This preference is available only through the administrative management utilities.
Allow "Refresh Cache" option via system tray	*Yes*/*No*/*Default*	If this option is set to *Yes*, the *Refresh Cache* option is not displayed and accessible in the notification area icon. If this option is set to *No* or *Default*, the *Refresh Cache* option is displayed in the notification area icon. The default value is *No*. This preference is available only through the administrative management utilities.
Allow "Work Offline" option via system tray	*Yes*/*No*/*Default*	If this option is set to *No* or *Default*, the *Work Offline* option is displayed in the notification area icon. If this option is set to *Yes*, the *Work Offline* options is not displayed in the notification area icon. The default value is *No*. This preference is available only through the administrative management utilities.

Preference	Value	Description
Allow application definition to be modified by users	*Yes*/*No*/*Default*	If this option is set to *Yes* or *Default*, end user can view and modify their application definitions. If this option is set to *No*, the end user cannot change their application definitions. The default option is *Yes*. **NOTE:** Disabling this preference does not disable the users from creating new applications through the wizards. This preference is available only through the administrative management utilities.
Allow application definition to be viewed by users	*Yes*/*No*/*Default*	If this option is set to *Yes* or *Default*, users can view the application definition. If this option is set to *No*, users cannot view the application definition. The default value is *Yes*. This preference is available only through the administrative management utilities.
Allow credentials to be deleted by users through the GUI	*Yes*/*No*/*Default*	If this option is set to *Yes* or *Default*, users can delete their credentials through the GUI. If this option is set to *No*, users cannot delete their credentials. The default value is *Yes*. This preference is available only through the administrative management utilities.
Allow credentials to be modified by users through the GUI	*Yes*/*No*/*Default*	If this option is set to *Yes* or *Default*, users can modify their credentials through the GUI. If this option is set to *No*, users cannot modify their credentials through the GUI. The default value is *Yes*. This preference is available only through the administrative management utilities.
Allow users to (de) activate SSO via system tray	*Yes*/*No*/*Default*	If this option is set to *Yes* or *Default*, users can switch between active and inactive modes of Novell SecureLogin. If this option is set to *No*, Novell SecureLogin is always active. User do not have the option to switch. The default value is *Yes*. This preference is available only through the administrative management utilities.

Preference	Value	Description
Allow users to backup/ restore	*Yes*/*No*/*Default*	If this option is set to *Yes* or *Default*, users can back up and restore their single sign-on information. If this option is set to *No*, users cannot back up and restore their single sign-on configuration. The default value is *Yes*. This preference is available only through the administrative management utilities.
Allow users to change passphrase	*Yes*/*No*/*Default*	If this option is set to *Yes* or *Default*, users can change their passphrase through the notification area icon. If this option is set to *No,* the *Change Passphrase* option is not displayed and users cannot change their passphrase through the notification area icon. The default value is *Yes*. This preference is available only through the administrative management utilities.
Allow users to modify names of Applications and Logins	*Yes*/*No*/*Default*	If this option is set to *Default*, Novell SecureLogin behaves as if it is set to *Yes*. The default value is *Yes*. This preference is available only through the administrative management utilities.
Allow users to view and change Preferences	*Yes*/*No*/*Default*	If this option is set to *Yes* or *Default*, users can view and change their preferences. If this option is set to *No*, users cannot view and change their preferences. The default value is *Yes*. **NOTE:** We recommend that you create a separate ou to ensure that they are not adversely affected by the general user configuration preferences at the ou level. This preference is available only through the administrative management utilities.

Preference	Value	Description
Allow users to view and modify API preferences	*Yes*/*No*/*Default*	The API preference defines the following options for users to: ◆ Enter an API licence key(s). ◆ Provide API access. If this option is set to *Yes* or *Default* users can view and modify the API preference. If this option is set to *No*, users cannot view and modify the API preference. The default value is *Yes*. This preference is available only through the administrative management utilities.
Allow users to view passwords	*Yes*/*Yes, per applications*/*No*/*Default*	If this option is set to *Yes* or *Default*, users can view their passwords. If this option is set to *No*, users cannot view their passwords. If this option is set to *Yes, per application*, users can view their passwords for only specific applications. The default value is *Yes*. **NOTE:** Allowing users to view their passwords gives them an opportunity to view and record passwords if they need to reset the Novell SecureLogin configuration. This preference is available only through the administrative management utilities.
Change the cache refresh interval (in minutes)	*5*	This preference defines the time in minutes of the synchronization of the user data and directory on the local workstation. The default value is set to *5* minutes. However, depending on the network traffic and the number of users the interval can be set between 240 minutes and 480 minutes (four and eight hours). This preference is available in both the Personal Management utility and the administrative management utilities.
Container has priority over User	*Yes*/*No*/*Default*	If this option is set to *Yes*, the container settings has priority over user settings. If this option is set to *No* or *Default*, the container settings does not have priority over the user settings. The default value is *No*. This preference is available only through the administrative management utilities.

Preference	Value	Description
Detect incorrect passwords	*Yes*/*No*/*Default*	If this option is set to *Yes* or *Default*, incorrect passwords for Web applications are detected. If this option is set to *No*, incorrect passwords for Web applications are not detected. The default value is *Yes*. This preference is available in both the Personal Management utility and the administrative management utilities.
Disable single sign-on	*Yes*/*No*/*Default*	If this option is set to *Yes*, access to Novell SecureLogin is disabled. If this option is set to *No* or *Default*, access to Novell SecureLogin is enabled. The default value is *No*. This preference is available only through the administrative management utilities.
Display splash screen on startup	*Yes*/*No*/*Default*	If this option is set to *Yes* or *Default*, the splash screen appears when Novell SecureLogin startsup. If this option is set to *No*, the splash screen is hidden and users cannot see the splash screen when Novell SecureLogin startsup. The default value is *Yes*. This preference is available only through the administrative management utilities.
Display the system tray icon	*Yes*/*No*/*Default*	If this option is set to *Yes* or *Default*, the Novell SecureLogin icon appears on the notification area. If this option is set to *No*, the Novell SecureLogin icon does not appear on the notification area. The default value is *Yes*. **NOTE:** When the Novell SecureLogin is active, users can double-click the icon on the notification area to launch the Personal Management utility. When the Novell SecureLogin is inactive, user can start the Personal Management utility through *Start* > *Programs* > *Novell SecureLogin* > *Novell SecureLogin* This preference is available only through the administrative management utilities.

Preference	Value	Description
Enable cache file	*Yes*/*No*/*Default*	This options defines the enabling or disabling of the creation of a Novell SecureLogin cache file on the local workstation. The cache stores user configuration data: local and inherited. Set this option to *Yes* for mobile users. If this option is set to *No*, you cannot store files locally or you are have some conflicts with organizational security policy If this option is set to *Default*, Novell SecureLogin behaves as if it is set to *Yes*. The default value is *Yes*. This preference is available in both the Personal Management utility and the administrative management utilities.
Enable Logging to Novell Audit	*Yes*/*No*/*Default*	This preference defines the enabling or disabling of log events to be automatically sent to Novell Audit tool, NSure. The following ou or user objects are logged by NSure: ◆ Single sign-on client started ◆ Single sign-on client exited ◆ Single sign-on client activated by user ◆ Single sign-on client deactivated by user ◆ Password provided to an application by a script ◆ Password changed by the user in response to a change password command ◆ Password changed automatically in response to a change password command **NOTE:** The Novell Audit platform must be installed on the client with a registered application ID and schema file on the server. If this option is set to *Yes* or *Default*, logging to Novell Audit is enabled. If this option is set to *No*, logging to Novell Audit is disabled. The default value is *Yes*. This preference is available only through the administrative management utilities.

Preference	Value	Description
Enable the New Login Wizard on the system tray icon	*Yes*/*No*/*Default*	This preference defines the enabling or disabling the user's ability to create multiple logins for different accounts on the same application or server. If this option is set to *Yes* or *Default*, users can create multiple logins. If this option is set to *No,* users cannot create multiple logins. The default value is *Yes*. This preference is available only through the administrative management utilities.
Enforce passphrase use	*Yes*/*No*/*Default*	Enforces the user definition of a passphrase question and answer when Novell SecureLogin is launched. If this option is set to *Yes*, users must complete setting up their passphrase before they proceed with any other activity on the workstation. If this option is set to *No* or *Default*, users can postpone setting up the passphrase. The default value is *No*. This preference is available only through the administrative management utilities.
Enter API license key(s)	Specify API licence key(s)	Specify the API license key(s) provided by Novell SecureLogin to activate the API functionality for an application. **NOTE:** You can add more than one API license keys.
Password protect the system tray icon	*Yes*/*No*/*Default*	Restricts the users from accessing the Novell SecureLogin icon menu option (from the notification area) without their network login password. If this option is set to *Yes*, the Novell SecureLogin icon on the notification area is password protected. If this option is set to *No* or *Default*, the Novell SecureLogin icon on the notification area is not password protected. The default value is *No*. This preference is available in both the Personal Management utility and the administrative management utilities.

Preference	Value	Description
Provide API Access	*Yes*/*No*/*Default*	Enables or disables the API functionality. If this option is set to *Yes*, the API access is enabled. If this option is set to *No* or *Default*, the API access is disabled. The default value is *No*. This preference is available in both the Personal Management utility and the administrative management utilities.
Standalone distributed settings have priority over user's	*Yes*/*No*/*Default*	Allows or disallows the values of configuration settings made by user to take precedence over the configuration settings made after settings distribution. Use this preference in advanced standalone mode for overwriting locally applied scripts, settings, and credentials by centrally created credentials. Use this preference also for suers who receive the encrypted and signed settings. If this option is set to *Yes*, the standalone distributed settings have priority over user's settings. If this option is set to *No* or *Default*, the standalone distributed settings do not have priority over user's settings. The default value is *No*. This preference is available only in SecureLogin Manager.
Stop walking here	*Yes*/*No*/*Default*	Enables or disables the inheritance of settings from higher level containers or organizational units. If this option is set to *Yes*, the inheritance of settings from higher level containers or organizational units is disabled. Set the option to *Yes* during phased upgrades when higher levels might have a different version of Novell SecureLogin implemented. If this option is set to *No* or *Default*, the inheritance of settings from higher level containers or organizational units is enabled. The default value is *No*. This preference is available only through the administrative management utilities.

Table 3-2 *The Java Preferences Properties Table*

Preference	Value	Description
Add application prompts for Java applications	*Yes*/*No*/*Default*	If the preference is set to *Yes* or *Default*, as soon as Novell SecureLogin detects a Java application login page, it prompts the user to record it. If this option is set to *No*, this process never occurs, only Java predefined applications are prompted and supported The default value is *Yes*. This preference is available in both the Personal Management utility and the administrative management utilities.
Allow single sign-on to Java applications	*Yes*/*No*/*Default*	If the preference is set to *Yes* or *Default*, as soon as Novell SecureLogin detects a Java application login page, it prompts the user to enable it for single sign-on. If this option is set to *No*, Java applications are not enabled for single sign-on. The default value is *Yes*. This preference is available in both the Personal Management utility and the administrative management utilities.

Table 3-3 *The Security Preferences Properties Table*

Preference	Value	Description
Certificate selection criteria	Specify text to identify your certificate	Allows you to specify a text to uniquely identify a certificate (within searchable field only). This preference is available only through the administrative management utilities.
Current certificate	No certificate selected	Allows selecting a certificate other than the default certificate. This preference is available only through the administrative management utilities.

Preference	Value	Description
Enable passphrase security system	*Yes*/*No*/*Hidden*	Prevents a rouge administrator from accessing the user's single sign-on credentials because they are prompted for the user's passphrase answer it they try to reset the user's network password and start Novell SecureLogin. If this option is set to *Yes* or *Default*, the passphrase must be answered by the user. Consequently, user contribution and knowledge is required in specific configurations to start Novell SecureLogin. If this option is set to *Hidden*, the user is not requested to answer a passphrase question. It is automatically generated by SecureLogin according to the user's parameters. This process is then automatically used in the configuration where a passphrase is required. If this option is set to *No*, the passphrase system is absent. Consequently, there is no backup process to store the user key. If the primary key is lost, Novell SecureLogin cannot be used by this user. The default value is *Yes*. **NOTE:** The *Enable passphrase security system* preference is supported only with the datastore version 6.0. The *Disable passphrase security system* preference applicable for datastore version 3.5 is removed and is no longer supported. If you are using this preference with datastore version 3.5, you must upgrade the datastore version 6.0 to use the *Enable passphrase security system* preference. This preference is available only through the administrative management utilities.

Preference	Value	Description
Lost card scenario	*Allow passphrase*/ *Require smart card*	Determines how Novell SecureLogin handles a user forgetting, losing or damaging their smart card. The Lost card option can only be used if, and only if, the Enable passphrase security system option is set to *Yes* or *Hidden* and *Use smart card to encrypt single sign-on data* is set to one of the smart card values. If this option is set to *Allow passphrase* or *Default*, the passphrase functions as a secondary key. If the smart card is not available, the passphrase is required in online mode to retrieve credentials from the directory. If this option is set to *Require smart card*, then there is no way to retrieve the credentials. The default value is *Allow passphrase.* **NOTE:** This preference is not available to users who have not upgraded their datastore to version 6.0. This preference is available only through the administrative management utilities.
Require Smart Card is present for SSO and administration operations	*Yes*/*No*/*Default*	This preference requires that a smart card must be accessible by SecureLogin each time a single sign-on operation is performed by an end user operation or administration operation. If this preference is set, SecureLogin cannot start without the smart card. As soon as the smart card is removed, SecureLogin is locked. By default, this preference is not set. If this option is set to *Yes*, Novell SecureLogin cannot start without the smart card. As soon as the smart card is removed, Novell SecureLogin is locked. If this option is set to *No* or *Default*, Novell SecureLogin can start without the smart card. The default value is *No*. **NOTE:** ◆If the *Lost card scenario* is set to *Allow passphrase*, the *Require smart Card is present for SSO and administration operations* preference is dimmed. ◆ If the *Lost card scenario* is set to *Require smart card*, then the *Require smart Card is present for SSO and administration operations* preference is available and behaves as if set to *No*. ◆ This preference is not available to users who have not upgraded their datastore to version 6.0. This preference is available only through the administrative management utilities.

Preference	Value	Description
Store credentials on smart card	*Yes*/*No*/*Default*	Allows you to store application credentials only on smart card. If this option is set to *Yes*, all credentials are stored in the PIN-protected area of a smart card instead of being encrypted in the cache file. If this option is set to *No* or *Default*, credentials are not stored in the PIN-protected area of a smart card. Scripts, settings, and policies are stored in the user's local cache, which is a mandatory preference for using smart cards. The default value is *No*. This preference is not available to users who have not upgraded their datastore to version 6.0. This preference is available only through the administrative management utilities.
Use AES for SSO data encryption	*Yes*/*No*	This option is defined to change the data encryption mode. This option is not available prior to version 6.0 of Novell SecureLogin. If the preference is set to *Yes* or *Default*, you can use AES instead of Triple DES for encrypting single sign-on data. If the preference is set to *No*, you cannot use AES instead of Triple DES for encrypting single sign-on data. The default value is *Yes*. This preference is available only through the administrative management utilities.
Use enhanced protection by default	*Yes*/*No*/*Default*	This setting is only relevant in a Novell environment; it concerns the SecretStore protection. If this option is set to *Yes* or *Default*, then a password protection is added. If this option is set to *No*, a password protection is not added. The default value is *Yes*. This preference is not available to users who have not upgraded their datastore to version 6.0. For details, see the SecretStore documentation. (http://www.novell.com/documentation/secretstore34/index.html) This preference is available only through the administrative management utilities.

Preference	Value	Description
Use smart card to encrypt SSO data	*No*/*PKI credentials*/ *Key stored on smart card*	Allows PKI credentials or a self-generated key to be created as the encryption source to encrypt the single sign-on data in the directory. If this preference is set to *No* or *Default*, all other smart card options are dimmed. If this preference is set to *PKI credentials*, single sign-on data is encrypted using the user's PKI credentials. Single sign-on data stored in the Directory and in the offline cache (if enabled) is encrypted using the public key from the selected certificate and he private key (stored on a PIN-protected smart card) is used for decryption. If this preference is set to *Key stored on smart card*, single sign-on data is encrypted using a randomly generated symmetric key that is stored on the user's smart card. This key is used to encrypt and decrypt single sign-on data stored in the Directory and in the offline cache (if enabled). The default preference is *No*. This preference is available only through the administrative management utilities.

Table 3-4 *The Web Preferences Properties Table*

Preference	Value	Description
Add application prompts for Internet Explorer	*Yes*/*No*/*Default*	The default value is *Yes*. This preference is available in both the Personal Management utility and the administrative management utilities.
Add application prompts for Mozilla Firefox	*Yes*/*No*/*Default*	The default value is *Yes*. This preference is available in both the Personal Management utility and the administrative management utilities.
Allow single sign-on to Internet Explorer	*Yes*/*No*/*Default*	The default value is *Yes*. This preference is available in both the Personal Management utility and the administrative management utilities.
Allow single sign-on Mozilla Firefox	*Yes*/*No*/*Default*	The default value is *Yes*. This preference is available in both the Personal Management utility and the administrative management utilities.

Preference	Value	Description
Allow single sign-on to Netscape	*Yes*/*No*/*Default*	The default value is *Yes*. This preference is available in both the Personal Management utility and the administrative management utilities.

Table 3-5 *The Windows Preferences Properties Table*

Preference	Value	Description
Add application prompts for Windows applications	*Yes*/*No*/*Default*	The default value is *Yes*. This preference is available in both the Personal Management utility and the administrative management utilities.
Allow single sign-on to Windows applications	*Yes*/*No*/*Default*	The default value is *Yes*. This preference is available in both the Personal Management utility and the administrative management utilities.

Managing Passphrases

4

This section provides information on the following:

4.1 About Passphrases

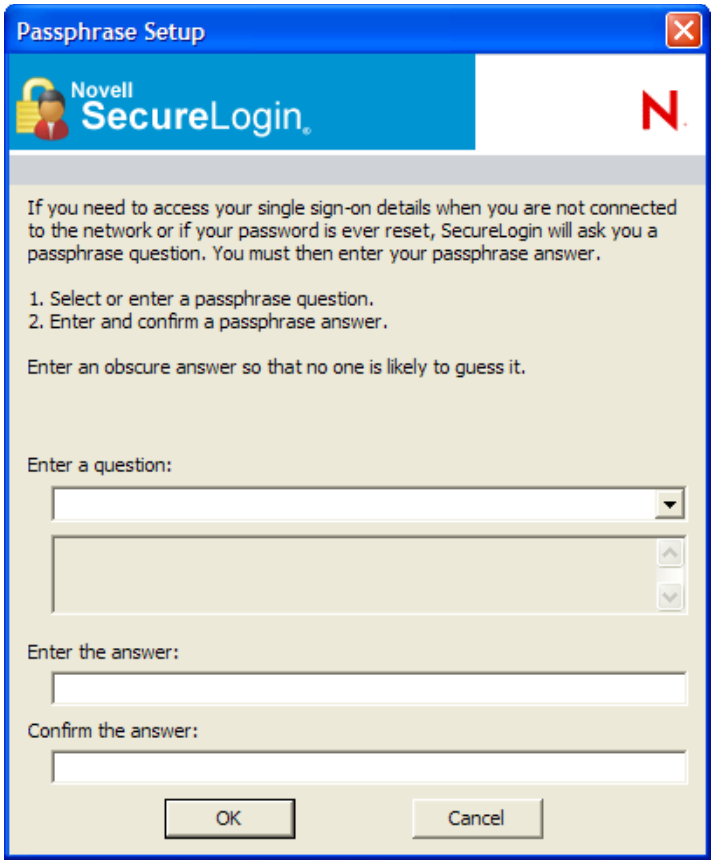
Passphrases are an important security component in the implementation of Novell® SecureLogin. Passphrases are unique question and answer combinations created to verify and authenticate the individual. In a directory environment, you can create passphrase questions for users to select a question and provide an answer for it. You can also permit users to select or provide questions and answers.

Passphrases protect user credentials from unauthorized use. For example, in a Microsoft Active Directory* environment, you can potentially log in to the network by resetting the user’s network password.

However, this cannot happen you are using Novell SecureLogin. If someone other than the actual users tries to reset the network password, Novell SecureLogin triggers the passphrase question. The user must provide the correct answer before successfully logging in. Even an administrator cannot access the user’s single sign-on-enabled applications without knowing the user’s passphrase answer.

When Novell SecureLogin is launched for the first time on the user’s workstation, the Passphrase Setup dialog box is displayed.

Figure 4-1 *Passphrase Setup Dialog Box*



Passphrase Authentication

Passphrases are used to authenticate when:

- A user is working either remotely or offline in an eDirectory or non-Microsoft Active Directory LDAP environment.
- Someone other than the user has reset the actual user's network password.

Benefits of Passphrases

Some of the benefits of using passphrase include:

- An individual cannot access a user's credentials by resetting the network password.
- Passphrases can be used in conjunction with SecureLogin Self-Service Password Reset, which enables users to reset their network password after answering the passphrase question.
- You can use this functionality to disable access to user credentials if the computer is stolen.

NOTE: You can disable the passphrase security system, but it also removes the features mentioned in the preceding section.

4.2 Creating a Passphrase Question

As an administrator, you can do the following:

- Create one or more passphrase questions for users to select.
- Enable users to create their own passphrase question and answer.
- Set up a combination of both.

To create a passphrase question:

1. Access the Administrative Management utility of Novell SecureLogin.

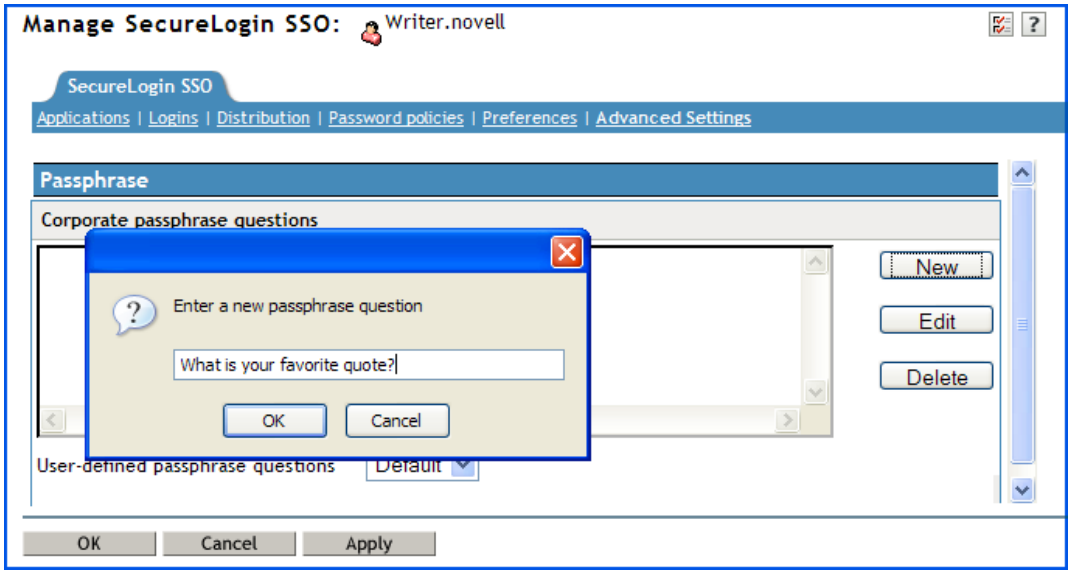
   For information on accessing the Administrative Management utility, see Section 1.2, "Starting the Administrative Management Utilities," on page 14 and, or, Section 1.3, "Accessing the Single Sign-On Plug-In Through iManager," on page 15.

2. Click *Advanced Setting*. The advanced setting options are deployed.

   By default, the *User-defined passphrase questions* is selected. Deselect this option if you do not want users to create their own passphrase question and answer.

3. Click *New*.

4. In the Enter a new passphrase question dialog box, provide your passphrase question.



5. Click *OK*. The question you provided is displayed in the *Corporate passphrase questions* field.

   This passphrase question is displayed to all users associated with the selected object.

6. Repeat the Steps 3 to Step 5 to create additional passphrases.

IMPORTANT: Make sure you click *OK* after you have created the passphrase question to save the changes and exit the page.

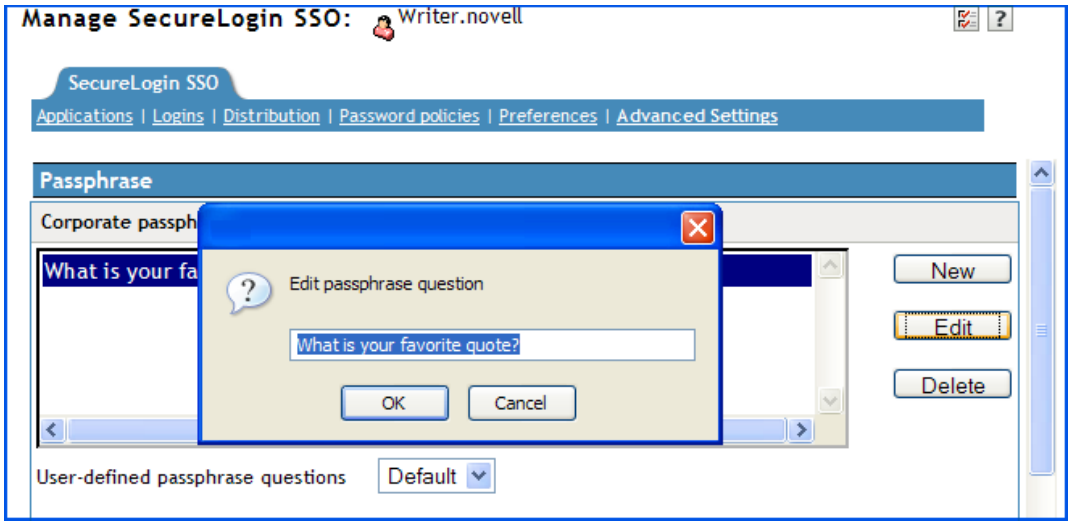
The passphrase answer is specified by the user when he or she sets up the passphrase question and answer. Ideally, passphrase answers must contain a minimum of six characters. However, you can change the policy to suit your security requirement.

We recommend that you do not apply strict policies to passphrase answers as it make them harder to remember. Instead, we recommend you use a multivalue question, such as `What is you driving license number plus your age?` and set a passphrase policy based on that.

4.3 Editing a Passphrase Question

1. Access the Administrative Management utility of SecureLogin.

   For more information on how to access the Administrative Management utility, see Section 1.2, "Starting the Administrative Management Utilities," on page 14 and Section 1.3, "Accessing the Single Sign-On Plug-In Through iManager," on page 15.
2. Click *Advanced Settings*. The Advanced Settings options are displayed.
3. In the *Corporate passphrase questions* box, select the passphrase question you want to edit.
4. Click *Edit*.



5. Make the required changes, then click *OK*. The passphrase question is updated with the changes.

IMPORTANT: Make sure that you click *OK* after you have created the passphrase question to save the changes and exit the page.

4.4 Deleting a Passphrase Question

To delete an existing passphrase question:

1. Access the Administrative Management utility of SecureLogin.

   For more information on how to access the Administrative Management utility, see Section 1.2, "Starting the Administrative Management Utilities," on page 14 and Section 1.3, "Accessing the Single Sign-On Plug-In Through iManager," on page 15.
2. Click *Advanced Settings*. The Advanced Settings options are displayed.
3. In the *Corporate passphrase questions* box, select the passphrase question you want to delete.
4. Click *Delete.* The selected passphrase question is deleted.

4.5 Re-setting a Passphrase Answer

If a user forgets the passphrase answer, you must reset the user's Novell SecureLogin configuration to ensure that the user's data is secure. This deletes all user-specific information, including usernames and passwords.

For more information on re-setting user data, see Section 2.3, "Deleting or Re-setting User Data," on page 24.

IMPORTANT: When you set up a user's passphrase question and answer policies, we recommend that you keep them simple so that the user can easily remember the answer.

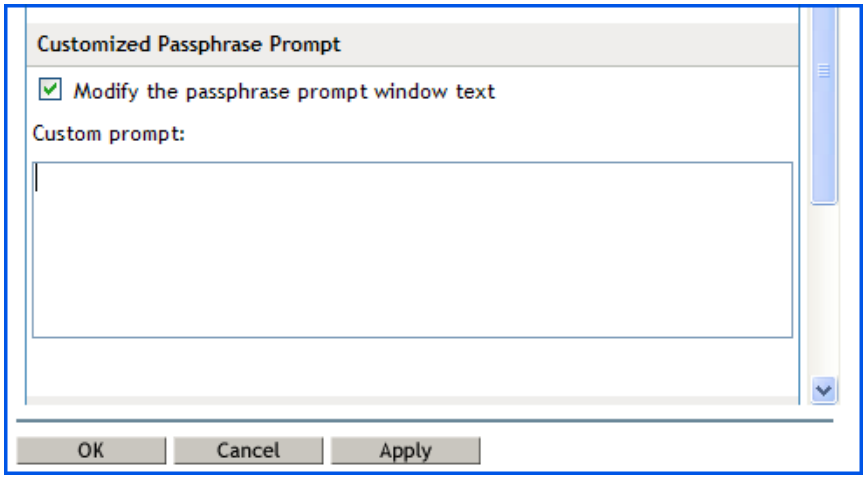
4.6 Changing the Passphrase Prompt

You can change the passphrase prompt that users see in the Passphrase Setup dialog box the first time they log in.

1. Access the Administrative Management utility of SecureLogin.

   For more information on how to access the Administrative Management utility, see Section 1.2, "Starting the Administrative Management Utilities," on page 14 and Section 1.3, "Accessing the Single Sign-On Plug-In Through iManager," on page 15.

2. Click *Advanced Settings*. The Advanced Settings options are displayed.
3. Under *Customized Passphrase Prompt*, select the *Modify the passphrase prompt window text* check box. The *Custom prompt* is now active.

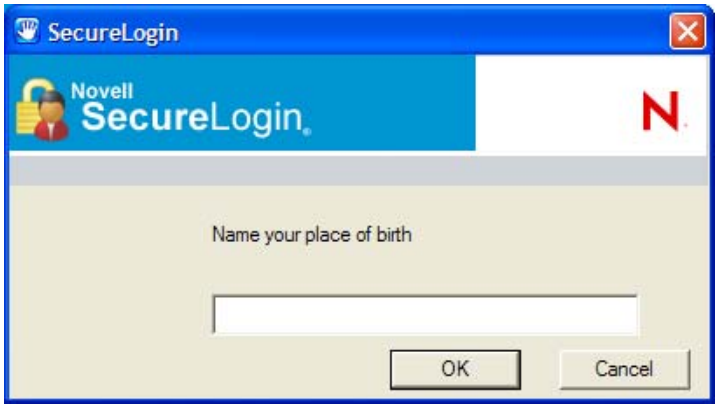


4. Specify the new prompt.
5. Click *OK* to save the changes and close the Administrative Management utility. Log in as a new user to view the customized prompt.

4.7 Changing a Passphrase

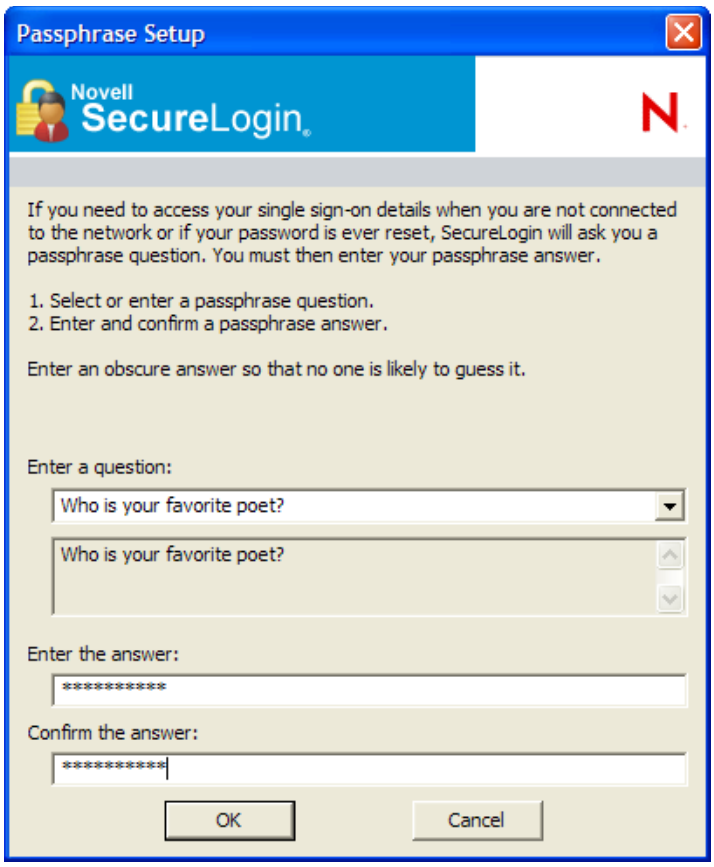
User can change their passphrase answer depending on how you configure Novell SecureLogin.

1. Right-click in the notification area, then select *Advanced* > *Change Passphrase*. The Passphrase dialog box is displayed.



2. Specify the passphrase answer in the field.
3. Click *OK*. The Passphrase Setup dialog box is displayed.
4. In the *Enter a question* field, select or specify a passphrase question.
5. In the *Enter the answer* field, specify the new passphrase answer.
6. In the *Confirm the answer* field, retype the new passphrase answer.

7 Click *OK*.



NOTE: Users who do not have access to the Novell SecureLogin icon cannot change their passphrases. You can enable access to the icon temporarily to allow the user to change the passphrase.

Managing Passphrase Policies

5

5.1 About Passphrase Policies

A passphrase is an integral part of the security architecture of SecureLogin. It can be used to secure single sign-on data when a user authenticates to applications.

You can set passphrase policies in the *Passphrase Policy properties* table of the Administrative Management utility, the iManager single sign-on plug-in, or the Group Policy plug-in.

You can set a policy to restrict the format and content of passphrase answers, including length, whether numeric characters are required, and whether passphrases must be uppercase or lowercase.

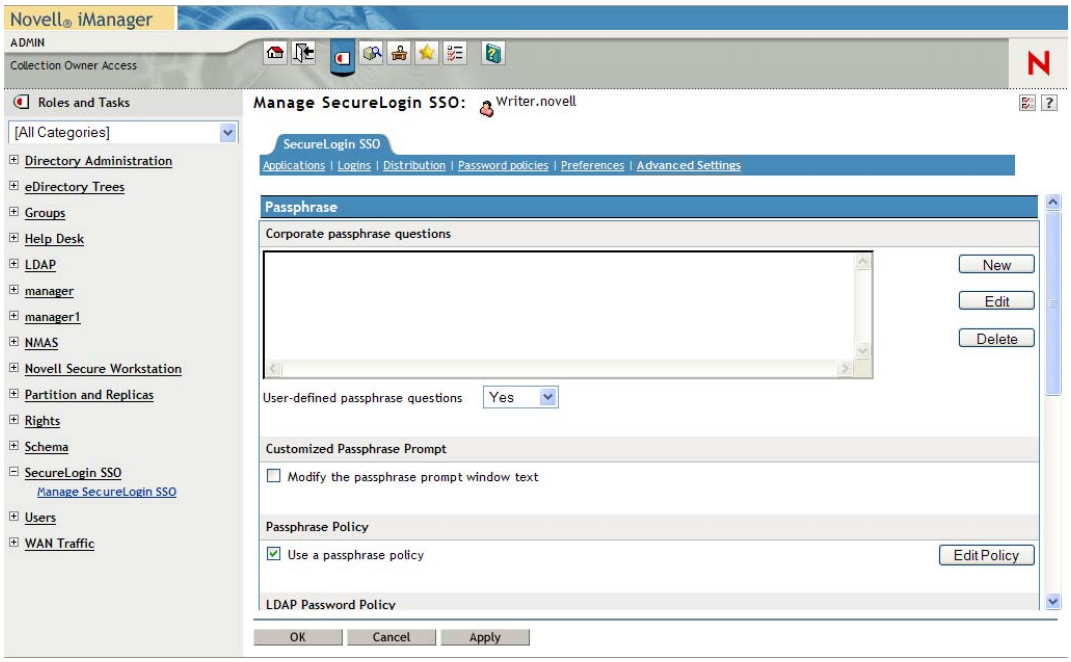
For example, you could set *Begin with an uppercase character* to *Yes* and *Maximum uppercase characters* to *1*, and *Prohibit characters* to *Disallow spaces*. You could also require all passphrase answers to start with uppercase and have the rest of the characters as lowercase.

If you set a passphrase policy, the policy must be applicable to all passphrase questions. You cannot enforce a passphrase policy similar to the one explained in the proceeding paragraph and then include a passphrase question such as “What is your mobile phone number?” because this question does not contain a combination of uppercase and lowercase and meet the other requirements.

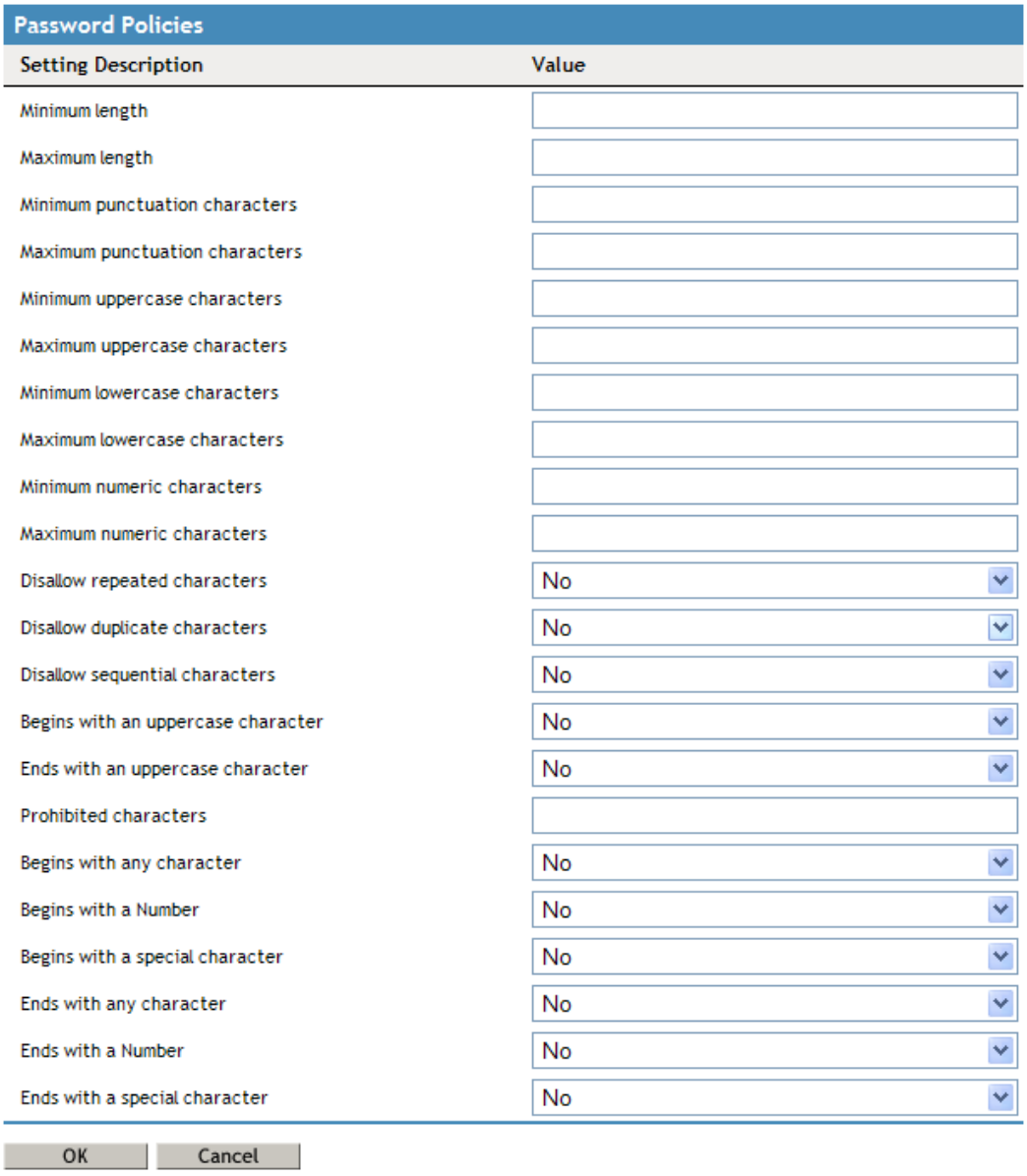
5.2 Changing a Passphrase Policy

1. Access the Administrative Management utility of Novell SecureLogin.

   For information on accessing the Administrative Management utility, see Section 1.2, “Starting the Administrative Management Utilities,” on page 14 and, or, Section 1.3, “Accessing the Single Sign-On Plug-In Through iManager,” on page 15.
2. Click *Advanced Settings*. The Advanced Settings options are displayed.
3. Select the *Use a passphrase policy* check box.



4 Click *Edit Policy.* The Passphrase Policy settings page is displayed.

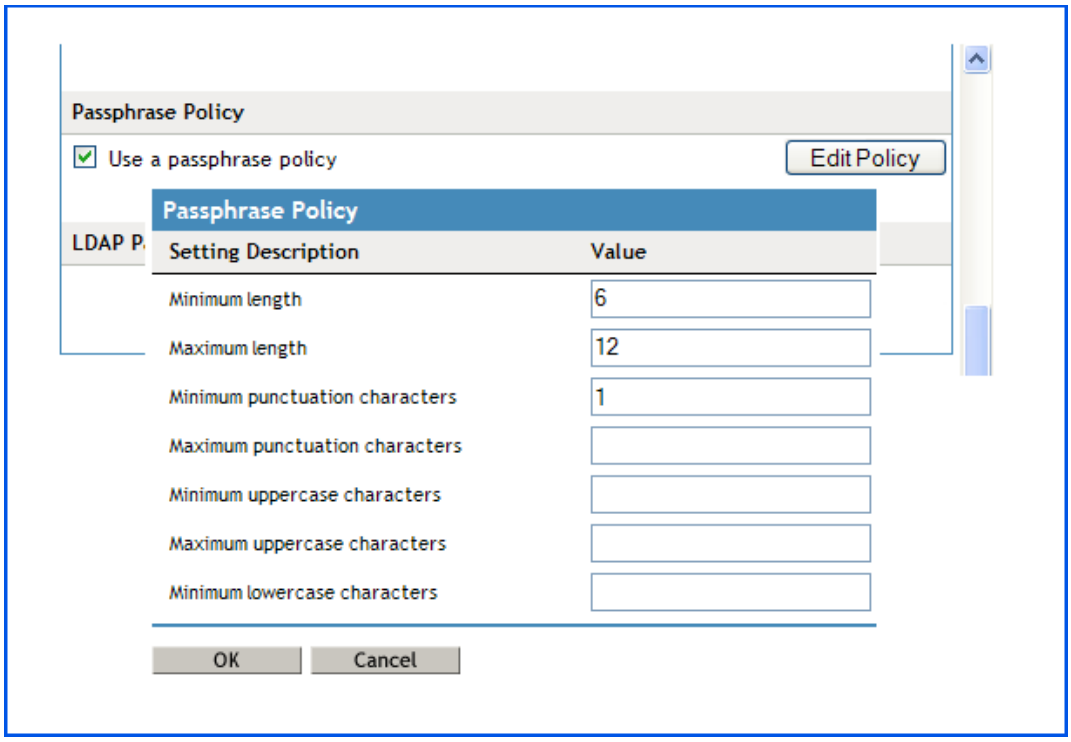


5 In the *Setting Description* column, click the policy rule you want to edit, then in the *Value* column, specify the required value.

For example, if you think that users might find it easier to remember basic rules for all passphrases instead of remembering exactly how they typed a passphrase when they created it, you could require all passphrases to contain a minimum of four characters and a maximum of 12 characters. Set *Minimum length* to `6` and set *Maximum length* to `12`.

By default, passphrase responses are required to contain a minimum of six characters. For security reasons, any passphrase policy you implement must also contain a minimum of six characters.

6 When you have finished setting the values in the table, click *OK*. The new values are added to the *Value* column.



The passphrase policy now applies to all users inheriting configuration from the selected object. You can change or disable it at any time.

5.3 Enabling the Passphrase Security System

This section contains information on the following:

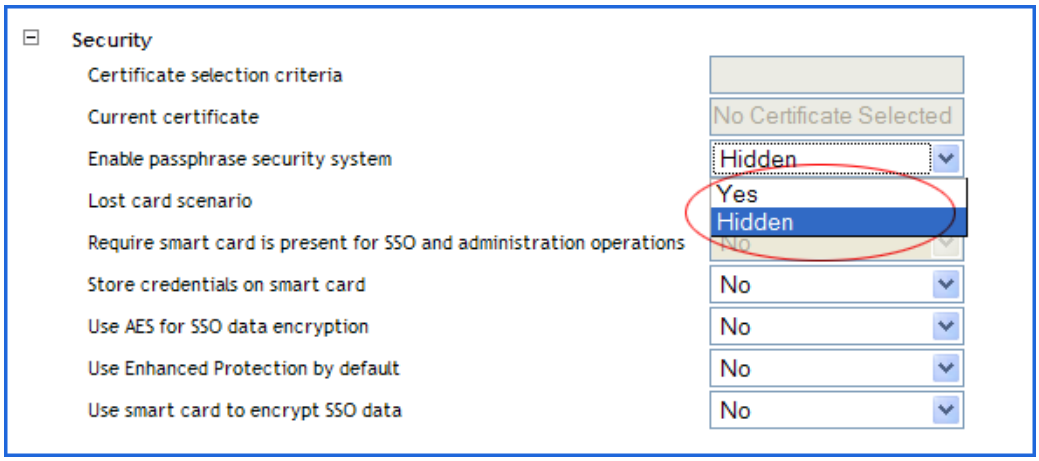
- Section 5.3.1, "Passphrases and Smart Cards," on page 58
- Section 5.3.2, "PKI Encryption and Passphrase Security," on page 59

The *Enable Passphrase Security System* option determines if users can use a passphrase to encrypt single sign-on data.

To view or modify this preference:

1. Access the Administrative Management utility of Novell SecureLogin.

   For information on accessing the Administrative Management utility, see Section 1.2, "Starting the Administrative Management Utilities," on page 14 and, or, Section 1.3, "Accessing the Single Sign-On Plug-In Through iManager," on page 15.
2. Click *Preferences*. The Preferences page is displayed.
3. Select *Security* > *Enable passphrase security system* and from the drop-down list, select either *Yes* or *Hidden*.



4 Click *Apply*.

5 Click *OK*.

You can set the *Enable Passphrase Security System preference* to *Yes* or *Hidden* depending on the enterprise security requirements.

If the *Enable Passphrase Security System* is set to *Yes,* (which is the default preference) the user is prompted to set the passphrase question and answer when Novell SecureLogin is launched for the first time.

If the *Enable Passphrase Security System* is set to *Hidden*, the user is not prompted to set the passphrase question and answer when Novell SecureLogin is launched for the first time.

WARNING: If you change the preference from *Hidden* to *Yes*, the users are prompted to re-specify their passphrase question and answer (after the initial set up). The users must specify thier the question and answer to proceed with the login.

The users are not indicated of the change you have made. So, we recommend that you do not change the preference.

You have two options, depending on what you specified.

- Users can create both the passphrase question and answer.
- You predefine a list of questions and answers, and the user selects from the list.

When users have set a passphrase, the application generates a random key, and a one-way hash of the passphrase answer encrypts this key. Later, the application key encrypts the new key. This key protects users' SecureLogin credentials and passwords so that even someone with Supervisor rights to the network and access to Microsoft Management Console (MMC) is unable to view a user's passwords to applications.

After the passphrase is set, every time that a user logs in to the network, Novell SecureLogin loads seamlessly.

Typically, the prompt to create a passphrase is never seen after the first login. However, if an administrator resets the user's directory or network, the next time SecureLogin launches, users must answer the passphrase question before SecureLogin continues. This prevents other users from changing the user's directory password, logging on as the user, obtaining access to the Novell SecureLogin data, and using it to run applications.

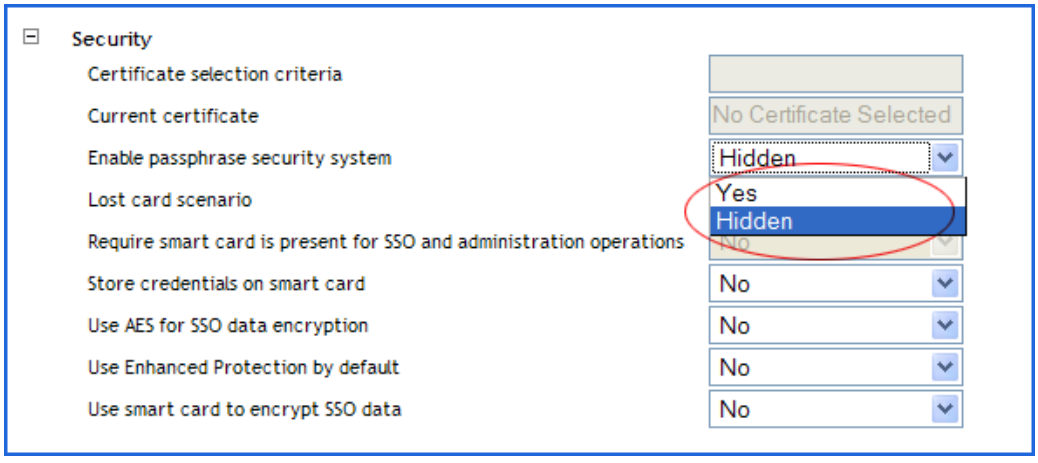
5.3.1 Passphrases and Smart Cards

You cannot toggle the *Enable Passphrase Security System* setting when the users forget their smart card unless they had previously set a passphrase or had it randomly generated using the *Hidden* option.

If users are required to authenticate to the network by using passwords, *Enable Passphrase Security System* must be set either to *Yes* or *Hidden*.

1. Access the Administrative Management utility of Novell SecureLogin.

   For information on accessing the Administrative Management utility, see Section 1.2, “Starting the Administrative Management Utilities,” on page 14 and, or, Section 1.3, “Accessing the Single Sign-On Plug-In Through iManager,” on page 15.
2. Click *Preferences*. The Preferences page is displayed.
3. Under *Security*, select either *Yes* or *Hidden* in the *Enable passphrase security passphrase* drop-down list.



4. Click *Apply*.
5. Click *OK*.

If you select *Yes*, users must select a passphrase question and answer when they log in to SecureLogin for the first time. When the passphrase system is enabled, users are prompted to answer their passphrase question if their password has been reset by the administrator.

NOTE: With the *Use smart card to encrypt SSO data* option selected (either *PKI credentials* or *Key generated on smart card*), you can use the passphrase to decrypt single sign-on data if the user’s smart card is damaged or lost.

This setting must be used in conjunction with the *Lost card scenario* preference set to *Allow passphrase* and *Store credentials on the smart card* preference set to *No*. You can toggle these preferences if the user's smart card is forgotten providing the user's passphrase has already been set. The user is prompted to answer the passphrase question before SecureLogin loads.

For more information, see Section 8.5, "Lost Card Scenarios," on page 95

If the *Hidden* preference is selected, users are not prompted to set a user-defined passphrase. A user key is generated automatically with any input from the user.

The *Enable Passphrase Security System* cannot be set to *No* unless *Use smart card to encrypt SSO data* is set to *PKI credentials*.

If users are required to authenticate to the network by using passwords, the *Enable passphrase security system* option must be set to *Yes* or *No* or *Hidden*.

IMPORTANT: With the passphrase security system set to *Hidden*, a directory administrator can reset a user's directory password, log in as the user, and access the user's single sign-on data because they are not prompted to answer a passphrase question.

5.3.2 PKI Encryption and Passphrase Security

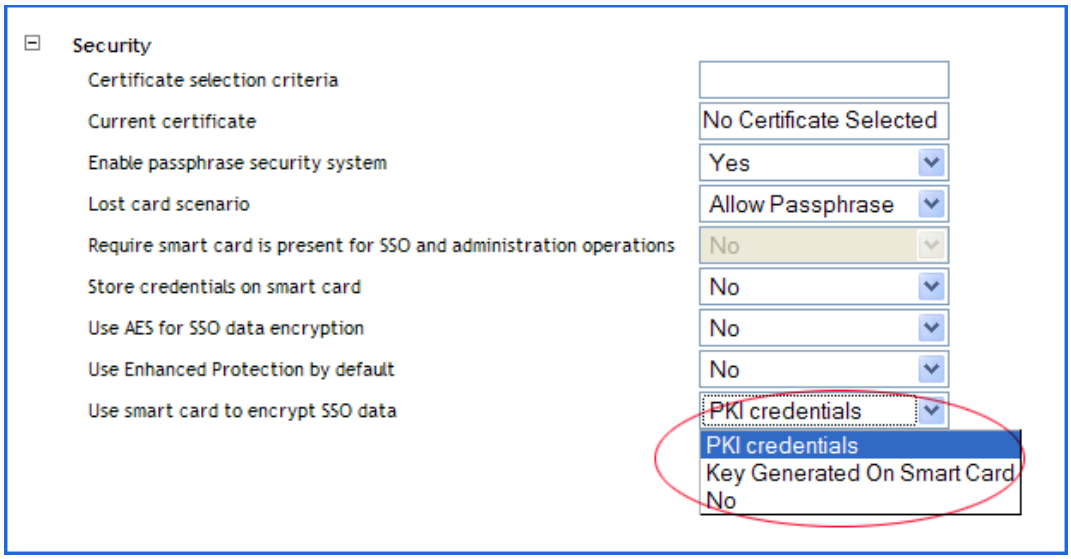
If the *Use smart card to encrypt SSO data* is set to *PKI credentials*, the user's single sign-on data is encrypted by using the public key from the selected certificate and the private key and stored on a PIN-protected container on the user's smart card. Both, the user's directory datastore and the local cache are now protected by the PKI credentials.

The single sign-on data can be encrypted by using the private key that is PIN-protected and stored on the user's smart card for added security. Only the user who has the physical possession of the smart card and knowledge of the PIN can decrypt the single sign-on data.

To set the *Use smart card to encrypt SSO data* preference:

1. Access the Administrative Management utility of Novell SecureLogin.

   For information on accessing the Administrative Management utility, see Section 1.2, "Starting the Administrative Management Utilities," on page 14 and, or, Section 1.3, "Accessing the Single Sign-On Plug-In Through iManager," on page 15.
2. Click *Preferences*. The Preferences page is displayed.
3. Select *Security* > *Use smart card to encrypt SSO data* and from the drop-down list, select either *PKI credentials* or *Key Generated On Smart Card* or *No*.



4 Click *Apply*.

5 Click *OK*.

If the *Use smart card to encrypt SSO data* is set to *PKI credentials*, the *Enable passphrase security system* can be optionally set to *No*.

If the *Use smart card to encrypt SSO data* is set to *No*, the user's passphrases are completely disabled and the user's smart card is always required to decrypt the single sign-on data.

IMPORTANT: If your enterprise chooses to disable the passphrase security system:

- You can still access a user's credentials by resetting the network password.
- The functions of using the passphrases in conjunction with SecureLogin Self Service Password Reset (SLSSPR) is disabled. The SecureLogin Self Service Password Reset enables a user to reset his or her network passwords after answering the passphrase questions.

The supported directory modes for disabling the passphrase security system are:

- Active Directory
- LDAP-compatible
- eDirectory (if SecretStore is used)

For detailed information on the likely scenarios that a user might experience in environments where the *Enable passphrase security system* option is set to *No*, see Section 5.5, "Passphrase Security System Scenarios," on page 61.

5.4 Checking the Passphrase Security System Status

1 On the notification area, right-click the Novell SecureLogin icon > *About*. The About dialog box is displayed.

The status appears next to the `Database Mode` and is listed as either `PP Enabled` or `PP Disabled`.

5.5 Passphrase Security System Scenarios

The information provided in this section describes the user experience in environments where the passphrase security system has been enabled and disabled.

Scenario 1: The passphrase security system is disabled in a previously enabled environment

When the passphrase security system is disabled in an environment where it was previously enabled, the following message appears to users when they log in for the first time, after the change.

Figure 5-1 *Passphrase Security Prompt*

If the user clicks *OK*, the disabling of the passphrase security system is approved and the user is prompted for the current password. The approval is complete when the user provides the password.

If the user click *Cancel*, the passphrase security system disabling is delayed and the user is prompted with the message until he or she click *OK* to approve the change.

NOTE: Users must answer the passphrase answer to prevent the administrators to toggle this preference and allow an unauthorized user access Novell SecureLogin.

Scenario 2: The passphrase security system is re-enabled in a previously disabled environment

If the passphrase security system is re-enabled, the Passphrase Setup dialog box is displayed (similar to when a user logs in for the first time after installing Novell SecureLogin.)

If the user clicks *OK*, the user resets the passphrase question and answer.

If the user clicks *Cancel*, there is a delay in enabling the passphrases for the user's workstation. The user is prompted at subsequent log ins until he or she specify the a passphrase question and answer.

Scenario 3: The passphrase security system is disabled and the user has changed his or her passwords (restrictions for moving user objects)

If you have disabled the passphrase security system and reset the user's password:

- In an LDAP-compatible and eDirectory (with SecretStore) modes, you cannot move the user object to another organizational unit until that user has logged in to Novell SecureLogin on his or her workstation. You must move the object back to its previous location to enable the user to run Novell SecureLogin.
- In an Active Directory mode, you can move the user object within the directory. However, copying is limited. If the user object is moved, you must move the object back to its previous location to enable the user to run Novell SecureLogin.

Scenario 4: Forgotten Passphrase

If a user forgets a his or her passphrase answer, the SecureLogin data, including their passphrase. You must delete the user's existing SecureLogin datastore.

After the datastore is deleted, the user's corporate applications, credentials, preferences, and user policies are permanently removed. You must then reset the user's corporate password before he or she can log in and reconfigure the applications by using Novell SecureLogin.

The next time Novell SecureLogin starts, he or she must manually log in. Novell SecureLogin then detects that a passphrase is not set and prompts the user to set up the passphrase before continuing. You can create a list of predefined list of passphrases questions.

After the user has set a new passphrase, he or she is required to re-enter the application usernames and passwords. It it is not done, an unauthorized could breach security by clearing the passphrase, entering a new passphrase, and accessing the actual user's credentials.

You might need to reset the user's application passwords as they might have forgotten them.

Managing Credentials

6

This section provides information on the following:

6.1 About Credentials

After you have created an application definition and activated it for single sign-on, the first time a user logs in, the user is prompted to provide credentials in a SecureLogin dialog box. SecureLogin then stores and associates these credentials with the application definition and uses it in subsequent logins.

You can display and manage these credentials in the *Logins* page of the Administrative Management utility and the *My Logins* pane of the Personal Management utility.

Because individual application requirements determine the credentials that users must enter when manually logging in, only those credentials are stored and remembered by SecureLogin. For example, if users have an application that only requires username and password, SecureLogin encrypts and stores the username and password for subsequent logins. Alternatively, some applications require the user to enter domain and database names, IP addresses, and select various options on Web pages. SecureLogin can handle all these on behalf of the user.

Credentials stored in a directory environment apply to all associated objects. For example, if users access an application located on a specific domain, and they are required to manually select or provide the domain address, then you can configure the domain as a credential in the *Logins* pane at the organizational unit level. This removes the requirement for users to manually provide the domain location when they log in. You can then change the domain at any time without notifying users.

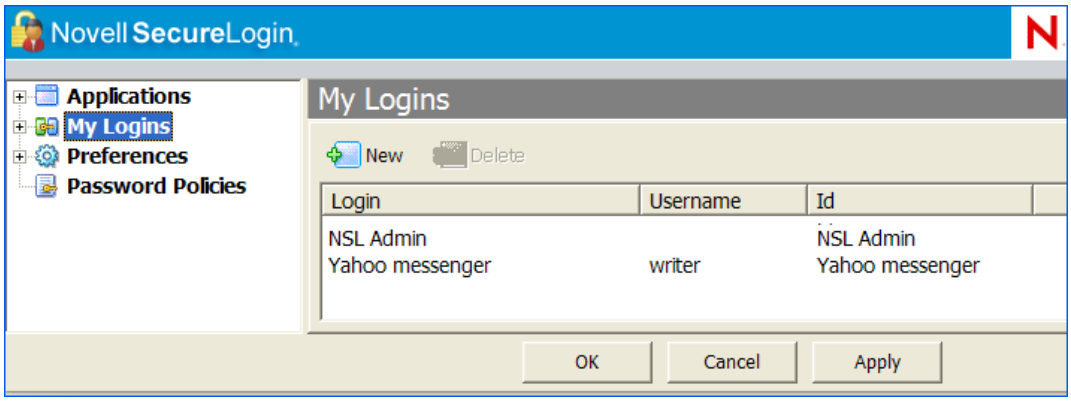
Application credentials such as e-mail, finance system, human resource system, and the travel system are typically stored for user objects and apply only to (and can be used by) the particular user. For example, John’s application credentials are encrypted and stored against John’s user object and only available to him. When he starts an application, SecureLogin retrieves, decrypts, and enters the credentials on behalf of John.

6.2 Creating a User Login and Credentials

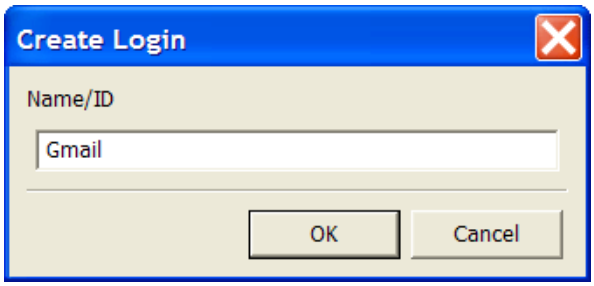
Logins and credentials are typically created automatically as part of the application definition, but you can manually create and edit them if required.

To create logins and credentials:

1. On the notification area, double-click the Novell SecureLogin the icon. The Personal Management utility is displayed.
2. Click *My Logins*. The existing Logins are displayed.



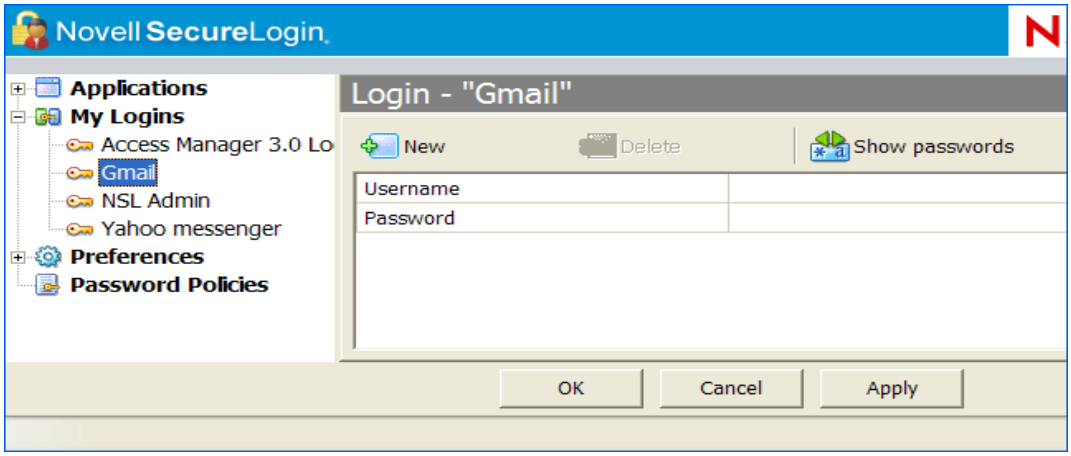
3 Click *New*. The Create Login dialog box is displayed.



4 In the *Name/Id* field, specify a Name/ID for the login.

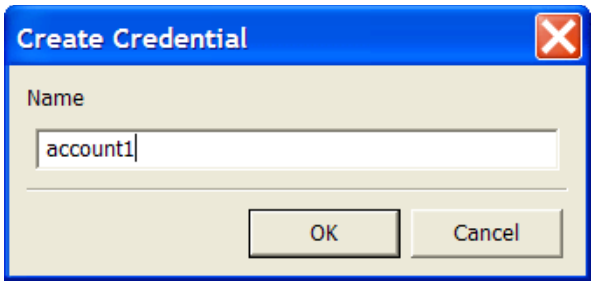
5 Click *OK*. The Login name/ID is added to the My Logins pane.

6 From the My Logins on the left pane, select the login you have created.

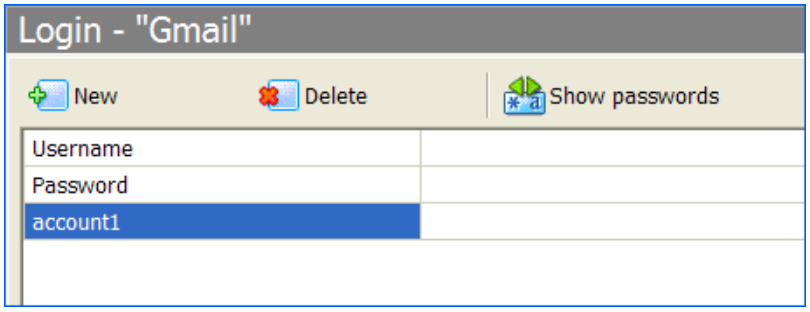


7 Click *New*. The Create Credential dialog box is displayed.

8 In the *Name* field, specify a name for the new credential.



9 Click *OK*. The new credential is added to the Login details.



10 In the Value column, specify a value for the credential.

11 Click *Apply*. The new credential variable and its value are displayed.

6.3 Linking a Login to an Application

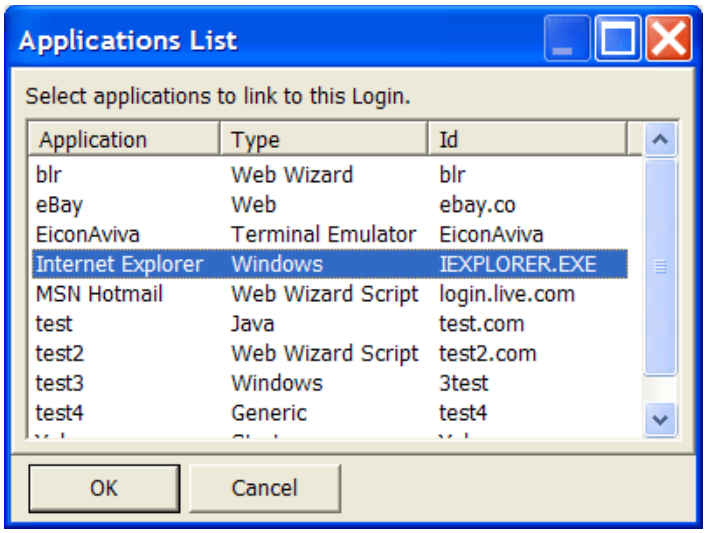
You can link a login to an application in the appropriate Login pane. For example, if users are logging in to Microsoft Outlook using a set of credentials and they are also logging in to Outlook Web Access, then they can share or link the credentials to the Web login application definition.

To link a login to an application:

1 In the notification area, double-click the Novell SecureLogin the icon. The Personal Management utility is displayed.

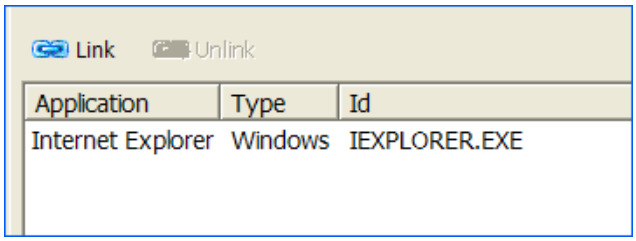
2 Click *My Login* and the login which you want to an application.

3 Click Link icon. The Applications List dialog box displays the list of enabled predefined applications and application definitions.



4 Select the application you want to link.

5 Click *OK*. The linked application is added.



6 Click *OK* to save changes and close the Personal Management utility.

6.4 Deleting Credentials

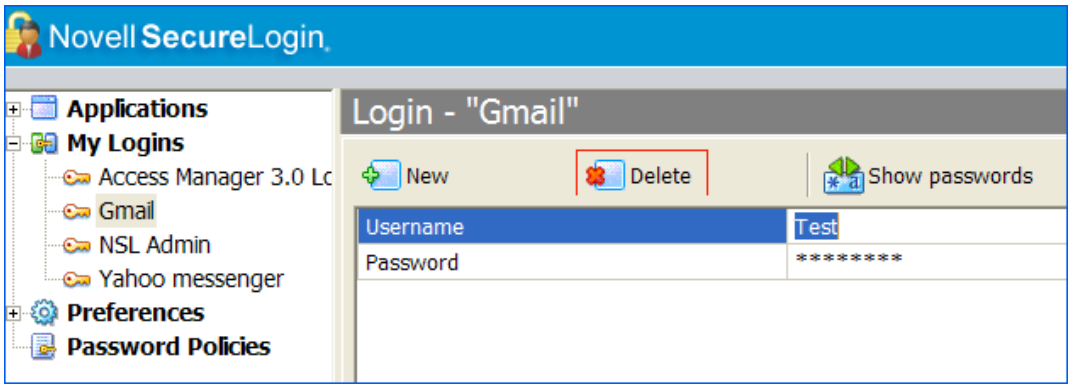
To delete log in credentials:

1 Open the Personal Management utility.

2 From the navigation on the left pane, select *My Logins*, then select the credential you want to delete.

3 On the right pane, click on the username of the credential you want to delete.

4 Click *Delete*.



5 Click on *Password*.

6 Click *Delete*.

The credential is deleted.

IMPORTANT: We recommend that you delete only two or three credential sets (usernames and passwords) in one session of the management utilities. This particularly important when the user's *Security* preference *Store credentials on smart card* is set to *Yes*.

Managing Password Policies

7

This section provides information on the following:

7.1 About Password Policies

SecureLogin provides password policy functionality to enable you to efficiently and effectively manage user passwords, in order to comply with your organization's security policies. You can create password policies at the container, OU, Group Policy, and user object level. Policies set at the container or organizational unit level are inherited by all associated directory objects. Password policies set at the user object level override all higher-level policies. Password policies are linked to application definitions through scripting and are not applied to directory objects. You can do this by creating a password policy in the *Password Policies* pane and then linking the policy to the application definition using the `RestrictVariable` command. However, the application definition is applied at the directory object.

Password policies are comprised of one or more password rules applicable to one or more single sign-on enabled applications and to specific directory objects. You can configure password policies in the *Password Policy Properties tables* of the Administrative Management utility, the iManager single sign-on plug-in, or the Group Policy plug-in.

SecureLogin remembers the passwords and can also handle password changes after they expire on the back-end application, for example, after 30 days or when users decide to change their password. The SecureLogin password management functionality includes the capability to set password expiry duration and generate passwords that comply with specified password policies.

NOTE: You can configure password change events by using SecureLogin’s wizards or through the application definition editor.

Password policies are typically created to match existing password policies. You should consult application owners before changing an existing password policy.

To determine the requirements and parameters of the password policy and the applications the password policy applies to, we recommend that you test complex policies on a test user account to ensure that they are viable.

7.2 Password Policy Properties

Organizations and applications often have rules about the content of passwords, including the required number and type of characters. The *Password Policy Properties* table helps you to create and enforce these password rules through a password policy, and apply this policy to one or more applications.

Table 7-1 *The Password Policy Properties Table*

Policy	Value To Be provided	Description
Minimum length	Whole number	Defines the minimum length of the password; that is, the number of characters required for the password.
Maximum length	Whole number	Defines the maximum length of the password; that is, the maximum number of characters allowed in password.
Minimum punctuation characters	Punctuation characters	Defines the minimum number of punctuation characters allowed in a password.
Maximum punctuation characters	Punctuation characters	Defines the maximum number of punctuation characters allowed in a password.
Minimum uppercase characters	Whole number	Defines the minimum number of uppercase characters allowed in a password.
Maximum uppercase characters	Whole number	Defines the maximum number of uppercase characters allowed in a password.
Minimum lowercase characters	Whole number	Defines the minimum number of lowercase characters allowed in a password.
Maximum lowercase characters	Whole number	Defines the maximum number of lowercase characters allowed in a password.
Minimum numeric characters	Whole number	Defines the minimum number of numeric characters allowed in a password.
Maximum numeric characters	Whole number	Defines the maximum number of numeric characters allowed in a password.

Policy	Value To Be provided	Description
Disallow repeat characters	*No*/*Yes*/*Yes, case insensitive*	Disallows the use of repeated characters, or the use of the same successive characters. If this option is set to *No*, characters can be repeated. This is the default value. If this option is set to *Yes*, same alphabetic characters in a different case are considered as different characters. For example, A and a are different. If this option is set to *Yes, case insensitive*, the successive use of the same alphabetic characters in a different case is not allowed.
Disallow duplicate characters	*No*/*Yes*/*Yes, case insensitive*	Disallows the use of the same non-successive characters. If this option is set to *No*, duplicate characters are allowed. This is the default value. If this option is set to *Yes*, the same alphabetic characters in a different case are considered as different characters. For example, A (uppercase) and a (lowercase) are different. If this option is set to *Yes, case insensitive*, duplication of the same alphabetic characters in a different case is not allowed.
Disallow sequential characters	*No*/*Yes*/*Yes, case insensitive*	Disallows the use of successive characters in an alphabetical order. If this option is set to *No*, sequential characters are allowed. This is the default value. If this option is set to *Yes*, sequential characters in a different case are considered as non-sequential. For example, a and b and non-sequential. If this option is set to *Yes, case insensitive*, sequential characters in different cases is disallowed.

Policy	Value To Be provided	Description
Begin with an uppercase character	*No*/*Yes*	Enforces the use of an uppercase alphabetic character as the beginning character of a password. The default value is *No*. If this option is set to *Yes*, all other policies that indicate that a password must begin with a particular character or in a specific manner are disabled. **IMPORTANT:** Only one type of character can be designated as the first value of a password.
End with an uppercase character	*No*/*Yes*	Enforces the use of an uppercase letter at the end of a password. The default value is *No*. If this option is set to *Yes*, all other policies that indicate that a password must end with a particular character or in a specific manner are disabled.
Prohibited characters	Keyboard characters	Defines a list of characters that cannot be used in a password. **NOTE:** There is no need of a separator in the list of prohibited characters. For example, @#$%&*
Begin with any Alpha character	*No*/*Yes*	Enforces the use of an alphabetic character at the beginning of a password. The default value is *No*. If this option is set to *Yes*, it automatically disables all other policies that specify what the first character of the password should be.
Begin with any number	*No*/*Yes*	Enforces the use of a numeric character as the first character of the password. The default value is *No*. If this option is set to *Yes*, it automatically disables all other policies that specify what the first character of the password should be.

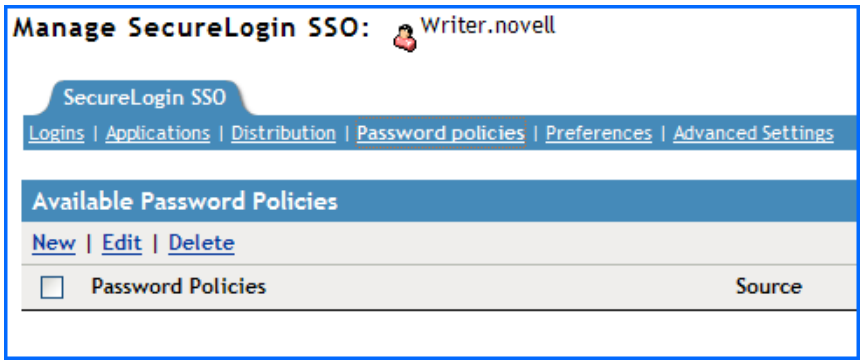
Policy	Value To Be provided	Description
Begin with any symbol	*No*/*Yes*	Enforces the use of a symbol character as the first character of the password. The default value is *No*. If this option is set to *Yes*, it automatically disables all other policies that specify what the first character of the password should be.
End with any Alpha character	*No*/*Yes*	Enforces the use of an alphabetic character as the last character of the password. The default value is *No*. If this option is set to *Yes*, it automatically disables all other policies that specify what the password should end with.
End with any number	*No*/*Yes*	Enforces the use of a numeric character as the last character of the password. The default value is *No*. If this option is set to *Yes*, it automatically disables all other policies that specify what the password should end with.
End with any symbol	*No*/*Yes*	Enforces the use of a symbol character as the last character of the password. The default value is *No*. If this option is set to *Yes*, it automatically disables all other policies that specify what the password should end with.

7.3 Creating a New Password Policy

To create a new password policy:

1. Access the Administration Management Utility.

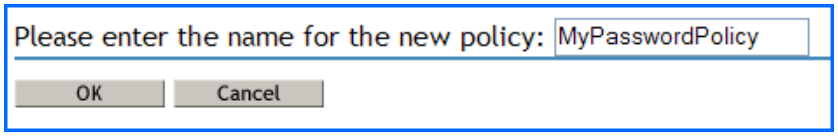
   For information on accessing the Administrative Management utility see, Section 1.2, “Starting the Administrative Management Utilities,” on page 14 and, or, Section 1.3, “Accessing the Single Sign-On Plug-In Through iManager,” on page 15.
2. Click *Password Policies*. The Password Policies page displayed.



3 Click *New*. The New Password Policy dialog box is displayed.

It is important to use a unique name for all logins, applications, and password policies. Password policies cannot have the same name as any other SecureLogin attribute. Organizations typically employ the naming convention ApplicationNamePwdPolicy, for example, LotusNotesPwdPolicy.

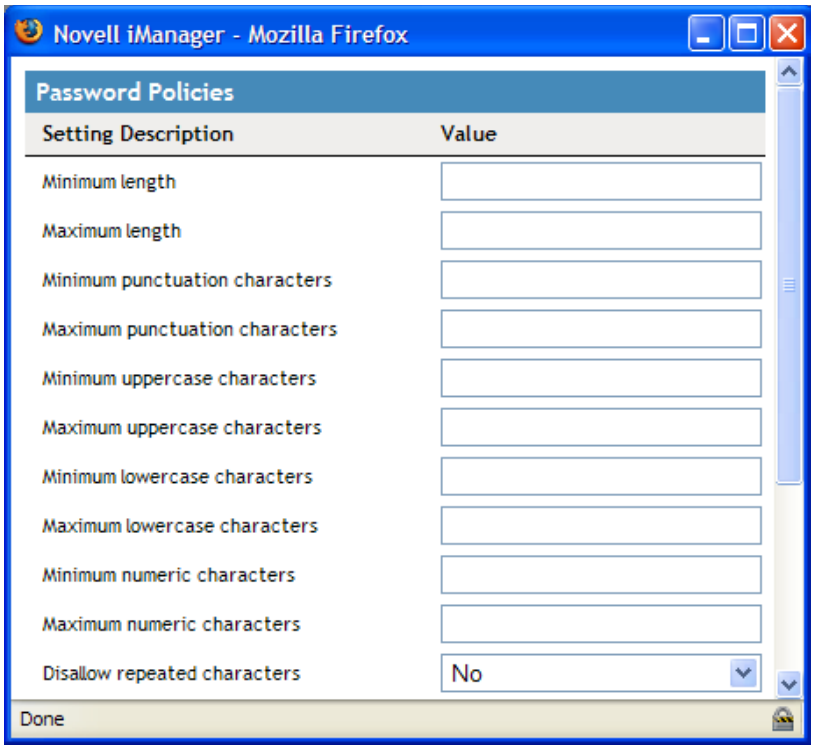
4 In the *Enter a name for the new password policy* field, specify a name for policy. The new policy is added under the Password Policies.



5 Click *OK*. The new password policy is added.

6 Click the new password policy. The Password policy properties table is displayed.

The table contains *Description* and *Value* columns. Most Policy rules are not enforced and do not have a default value. Values are either Yes, No, or a whole number.



7 In the *Description* column, locate the policy you want to change and then in the *Value* column, click the appropriate value from the drop-down list.

8 Click *Apply* to save changes.

9 Click *OK* to close the Administrative Management utility.

IMPORTANT: Password policies are linked to applications by using the SecureLogin application definition command `RestrictVariable`. You can use this command to apply password policies to one or more applications.

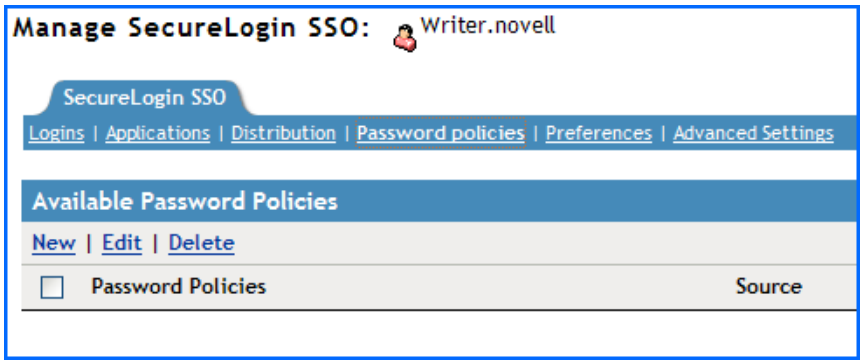
7.4 Changing a Password Policy

You can change a password policy by adjusting the parameters of each rule, or by having no parameters for a rule.

1 Access the Administration Management Utility.

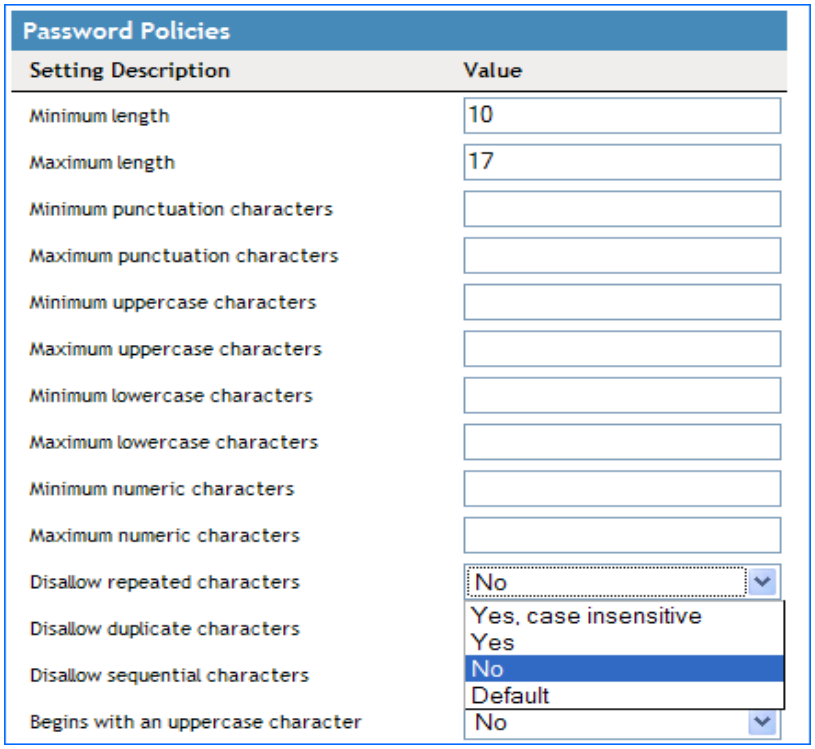
For information on accessing the Administrative Management utility, see Section 1.2, “Starting the Administrative Management Utilities,” on page 14 and, or, Section 1.3, “Accessing the Single Sign-On Plug-In Through iManager,” on page 15.

2 Click *Password Policies*. The Password policies page is displayed.



3 Click the password policy you want to change. The policy details are displayed.

4 In the Description column, locate the description you want to change, then in the Value column, select the appropriate value from the drop-down list.



5 Click *Apply* to save changes.

6 Click *OK* to close the Administrative Management utility.

7.5 Deleting a Password Policy

To delete a password policy:

1 Access the Administration Management Utility.

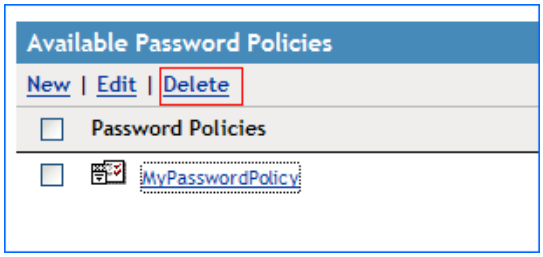
For information on accessing the Administrative Management utility see, Section 1.2, “Starting the Administrative Management Utilities,” on page 14 and, or, Section 1.3, “Accessing the Single Sign-On Plug-In Through iManager,” on page 15.

2 Click *Password Policies*. The password policies page is displayed.

3 Click the password policy that you want to delete.

4 Click *Delete*. The Password policy is deleted from the Password policies list.

You can also delete a Password policy by right-clicking the Password policy in the left or right pane of the Administrative Management utility and selecting the *Delete* option.



5 Click *Apply*.

6 Click *OK*.

7.6 Linking a Policy to an Application

You can link the password policies to applications by using the SecureLogin application definition command `RestrictVariable`. With this command, you can apply the password policies to one or more applications, as in the example below.

The following definition restricts the `$Password` variable to the Finance password policy. The user's password must match the policy when he or she saves the credentials. When the password requires changing, the application generates a new password based on the policy randomly because Random is included in the definition at ChangePassword.

```
# Set the Password to use the Finance Password Policy
RestrictVariable $Password FinancePwdPolicy
# Login Dialog Box
Dialog
   Class #32770
   Title "Login"
EndDialog
Type $Username #1001
Type $Password #1002
# Change Password Dialog Box
Dialog
   Class #32770
   Title "Change Password"
EndDialog
Type $Username #1015
Type $Password #1004
ChangePassword $Password Random
Type $Password #1005
Type $Password #1006
Click #1
```

The following example uses an application definition to restrict the `?NewPwd` variable to the Finance password policy. The user's current password ($Password) is saved and used when the application starts for the first time and prompts the user to enter the credentials.

When the password expires, the password policy is enforced on any new password. Through this means, you can enforce more strict password policies when you cannot guarantee all existing passwords meet the new policy.

```
# Set the Password to use the Finance Password Policy
RestrictVariable ?NewPwd FinancePwdPolicy
# Log on Dialog Box
Dialog
Class #32770
Title "Log on"
EndDialog
Type $Username #1001
Type $Password #1002
Click #1
# Change Password Dialog Box
Dialog
Class #32770
Title "Change Password"
EndDialog
Type $Username #1015
Type $Password #1004
ChangePassword ?NewPwd Random
Type ?NewPwd #1005
Type ?NewPwd #1006
Set $Password ?NewPwd
Click #1
```

Managing Smart Card Integration

8

This section provides information on the following:

8.1 How SecureLogin Uses Smart Cards

This section provides information on the following:

8.1.1 Prerequisites

The use of smart cards with Novell SecureLogin is based on the enterprise or corporate preference to allow user to utilize a smart card to log in and store their single sign-on data, or to encrypt their directory data by using a Public Key Infrastructure (PKI) token.

To enable the smart card support for Novell SecureLogin, the *Use smart card or cryptographic token* option must be selected during the installation regardless of your intended preferences for setting the security preference, *Require Smart Card is present for SSO and administration operations*.

IMPORTANT: Novell SecureLogin 6.1 SP1 only supports ActivClient*, Gemalto* (formerly Axalto), and AET SafeSign* smart card middleware.

8.1.2 Using Smart Card to Log In to Workstation

Novell SecureLogin allows a user to alternate their log in method by using both a smart card and their log in credentials.

However, a user can only log in by using both a smart card and password log in to access the SecureLogin credentials only if the *Use smart card or cryptographic token* option is selected during installation.

If the *Use smart card or cryptographic token* option is not selected during installation, a user attempting to access SecureLogin on the workstation is forced to log in with his or her username and network password.

Using Smart Card for an Initial Log In

When no smart card preferences are applied to the user, the *Enable passphrase security system preference* is set to *Yes*, and the user has not initially logged in by using his or her username and password, the following warning message is displayed.

Figure 8-1 *Smart Card Warning*

For security reasons, Novell SecureLogin requires a user to log in at least once by using the username and password before their smart card access is available.

When the *Enable passphrase security system* preference is set to *Hidden* and the smart card preferences are applied to the user, then the user can initially log in by using either the smart card or the username and password. The security warning is not displayed.

8.1.3 Storing Single Sign-on Credentials

SecureLogin uses a store-and-forward approach to single sign-on credentials, and records user IDs and passwords in this store. It is likely that many, if not all, of an individual user's passwords will be stored in this credential store. Given this architecture, the security of SecureLogin credential store is extremely important.

When a smart card is used in conjunction with SecureLogin, a number of new features can be optionally implemented to increase security. Some of them are:

- Using smart card to encrypt SecureLogin.
- Storing single sign-on credentials such as application usernames and passwords on the smart card.
- Typing single sign-on availability to the smart cards so only those who log in using a smart card are able to start and administer single sign-on.

SecureLogin uses a two-tier encryption process to secure users' sensitive credentials and information. All user passwords are encrypted with the user key, and all user data, including password fields, are encrypted with the master key.

The result is a two-tier encryption process where password values are encrypted twice: once with the user key and once with the master key, while all other data is encrypted once with master key.

Using SecureLogin in conjunction with a smart card provides an additional level of security because the key used to decrypt data is stored on the smart card, and authentication is through two-factor authentication: smart card and PIN. If the you select the option *Use smart card to encrypt SSO data* option, users must insert a smart card and enter a PIN for SecureLogin to load.

8.1.4 Authentication Methods

The following sections explain the strong authentication methods used in Novell SecureLogin.

Advanced Authentication

New Functionality in the AAVerify Command

This release of Novell SecureLogin enhances the `AAVerify` command functionality.

A predefined `AAVerify` application definition command (script) used to reauthenticate a user already exists in the previous releases of Novell SecureLogin.

For details of the `AAVerify` application definition command, see the *Novell SecureLogin 6.1 SP1 Application Definition Guide*.

The existing version of the `AAVerify` command relies on either SecureLogin Advanced Authentication (SLAA) or Novell Modular Authentication Services (NMAS) being deployed on the server of the backend to process any reauthentication calls.

The new `AAVerify` command was developed specifically provide a secure method to reauthenticate a user successfully before populating the Novell SecureLogin credentials for designated sensitive applications. In an enterprise or corporate environment, a sensitive application is one where a Novell SecureLogin application definition is applied calling for reauthentication.

To process the reauthentication request, the new `AAVerify` command now takes into account the method by which users are currently logged in, as well as their directory connectivity status.

If users have logged in with a username and password, they are prompted to reauthenticate by using the password, regardless of whether they are offline or online.

If users have logged in with a smart card, they are prompted to reauthenticate by using the original smart card PIN, regardless of whether they are offline or online.

The new `AAVerify` command is independent of SLAA or NMAS and can be used to enforce strong user-friendly re-authentication by using a smart card and PIN or password without installing SLAA or NMAS.

The new `AAVerify` command caters to a mixed environment where either of the following conditions exist:

- A user might log in to a number of workstations by using a combination of both smart card or password authentication
- A scenario where several users might log in to one workstation by either smart card or password authentication.

The New ?IsPin Variable

`?IsPin` is a new Novell SecureLogin variable available in Microsoft Active Directory mode only.

The `?IsPin` variable is automatically generated when a user logs in and stores, information based on whether the user has logged in to the workstation by using a smart card and PIN, or has logged in by using a password.

When the `?IsPin` variable is called from an application definition, it indicates the following:

- If the returned value is true, it means that the user has logged in by using a smart card, and only the PIN value is passed through to the Novell SecureLogin.
- If the returned value is false, it means that the user has logged with a password.

NOTE: The `?IsPin` variable is updated only at a login and is not updated at a screen unlock.

Supported Operating System Environment

The new `AAVerify` command functionality has been completely tested in the following systems:

- Microsoft Windows XP SP2
- Microsoft Windows 2000 SP4
- Microsoft Windows Vista (32-bit)

Supported Directory Environments

The Novell SecureLogin `AAVerify` command functionality is not currently supported in either LDAP or stand-alone mode.

Recommended Configuration

The *Use smart card or cryptographic token* option is normally based on your preference to have the Novell SecureLogin users utilize a smart card to store the single sign-on data or to encrypt their user's directory data by using a Public Key Infrastructure (PKI).

If you decide to allow users to log in to their workstations by using a smart card and reauthenticate against their smart card, then the *Use smart card or cryptographic token* option must be selected during the installation regardless of the option set for *Require smart card is present for SSO and administration operations*.

NOTE: We recommend that you use a smart card configuration policy to lock the screen on card removal to ensure that the smart card belongs to the currently logged-in user.

Example Application Definition

The following application definition shows how to call the `AAVerify` command based on the login method. It uses the Notepad application. After the Notepad application is started, the `AAVerify` command is invoked to prompt the user to reauthenticate, using the login method for the workstation.

```
Dialog
Class Notepad
EndDialog

OnException AAVerifyFailed Call AAVerifyFailed
OnException AAVerifyCancelled Call AAVerifyCancelled

If ?isPin Eq "true"
  AAVerify -method "smartcard" ?result
Else
  AAVerify -method "password" ?result
EndIf
ClearException AAVerifyFailed
ClearException AAVerifyCancelled

Type $username
Type \n
Type $password
Type \n
Sub AAVerifyFailed
   MessageBox "Reauthentication failed."
   EndScript
EndSub

Sub AAVerifyCancelled
MessageBox "Reauthentication cancelled."
EndScript
EndSub
## EndSection: "Login Window"
```

Reauthenticating a Predefined Web Application

If the new `AAVerify` command is used to reauthenticate a Web browser-based application or if the *Prompt for device authentication for this device* option is enabled for Web applications, then the predefined application definition for the Web browser must be applied for that particular user to avoid confusion when prompting for reauthentication.

One Time Password

The use of multiple passwords places high maintenance overheads on large enterprises. Users are frequently required to use and manage multiple passwords.

A one time password reduces the cost, particularly with regard to calls to the help desk to reset a forgotten password, or to ensure that all passwords are provisioned when a new user starts, or deleted when existing user leaves the organization.

SecureLogin integrates with ActivCard* one time password authentication functionality and provides you access to the `GenerateOTP` application definition command, which can be used to generate synchronous authentication and asynchronous authentication soft token support for smart card user authentication. For more information on one-time password see, Section 8.1.7, "One-Time Password," on page 84.

8.1.5 Network Authentication

Network authentication is the verification of a user's login credentials before granting access to a network or operating system. Users typically authenticate to a network using one of the following methods:

- Password
- Biometric device (fingerprint or iris scan)
- Smart card and PIN
- Token

When a user authenticates successfully and the operating system loads, SecureLogin starts and manages the login credentials to the user's single sign-on-enabled applications.

If you want to enforce biometric, smart card, or token authentication at the application (or transaction) level, SLAA or NMAS can be integrated with SecureLogin to prompt the user to re-authenticate before SecureLogin retrieves their credentials and logs in to single sign-on enabled applications.

Network authentication methods can also be integrated with SecureLogin to manage a user's Windows log in credentials. The authentication methods retrieves a user’s Windows username and password from the smart card and automatically enters this into the Windows Graphical Identification and Authorization (GINA) interface when the users enters a PIN.

8.1.6 Smart Card Application Re-Authentication

Stronger application re-authentication methods such as Secure Login Advanced Authentication and NMAS can also be integrated with SecureLogin to provide additional smart card and PIN re-authentication to single sign-on-enabled applications.

To do this, enable the *Prompt for device reauthentication for this application* option and configure the re-authentication method.

For information about configuring SecureLogin to re-authenticate an application, see Chapter 10, “Reauthenticating Applications,” on page 115.

8.1.7 One-Time Password

A one-time password is an authentication method specifically designed to avoid the security exposures inherited with traditional fixed and static password usage.

One-time passwords rely upon a predefined relationship between the user and an authenticating server. The encryption key is shared between the user's token generator (which can be a token or one-time password-enabled smart card) and the server, with each performing the pseudo-random code calculation at user login. If the codes match, the user is authenticated.

The main benefit of one-time password systems is that it is impossible for a password to be captured on the wire and replayed to the server. This is particularly important if a system does not encrypt the password when it is sent to the server, as is the case with many legacy mainframe systems.

SecureLogin uses an application definition (script) command to provide access to the `GenerateOTP` command, which can be used to generate synchronous and asynchronous authentication soft token support for smart card user authentication as well as hard token support for the Vasco* Digipass* token generator.

8.2 Installing SecureLogin for Smart Cards

This section contains information on installing SecureLogin for smart cards:

8.2.1 Client Setup

During the installation of SecureLogin, you can select the *Use smart card or cryptographic token* option to enable a SecureLogin user to utilize a smart card to store single sign-on data or to encrypt directory data by using a PKI token.

SecureLogin uses existing Novell smart card settings when they are detected (highly recommended) unless the you choose otherwise.

You can optionally select an alternative cryptographic service provider (Microsoft Crypto API) from a drop-down list for your preferred smart card or cryptographic token middleware and then select an appropriate smart card (PKCS#11) library file.

IMPORTANT: Manually configuring the third-party smart card PKCS #11 link library assumes a high level of understanding of the cryptographic service provider’s product. You are encouraged to use the ActivClient smart card support.

8.2.2 Server Side Administration Preferences

SecureLogin is a highly configurable and flexible product. Many options are available to the system administrator to implement and enforce corporate directory policy across an enterprise.

Corporate policies can include, but are not limited to, enabling strong application security, how single sign-on data is encrypted and stored, how password and passphrase policies are implemented and enforced, and how management procedures are set for a lost smart card.

If your company enforces strong security requirements, you should be fully aware of the implications of linking the use of single sign-on to a smart card and disabling the passphrase functionality.

Minimum Requirements

For general information about the minimum requirements for using smart cards with SecureLogin, refer to the *Novell SecureLogin 6.1 SP1 Installation Guide*.

Supported Configurations

- ActivClient 6.0 and 6.1

- Gemalto 5.3
- AET Safe Sign 2.3.0

NOTE: When deployed with ActivClient, SecureLogin automatically configures the cryptographic service provider and PKCS#11 dynamic link library file during installation.

If the appropriate version of PKCS#11 library file is not present during installation, SecureLogin installs without smart card support.

If ActivClient is installed after SecureLogin is installed, the registry key settings need to be changed manually to activate smart card support, uninstall or re-install SecureLogin.

Cryptographic Service Provider Middleware

ActivClient*, Gemalto*(formerly Axalto), and AET's SafeSign* smart card middleware and settings are automatically detected and selected for use during the installation of SecureLogin.

If the enterprise implementation of middleware does not use an ActivClient smart card, or you want to change the smart card or cryptographic token, then the appropriate cryptographic service provider middleware can be manually selected.

NOTE: Manually configuring a third-party smart card PKCS#11 link library assumes a high level of understanding of the cryptographic service provider's product. We recommend that you use ActiveClient smart card support with SecureLogin when ever possible.

8.3 Configuring SecureLogin for Smart Cards

No two organizations have the same environment and requirements, SecureLogin includes a number of options that determine SecureLogin's behavior, such as how single sign-on data is encrypted (that is, using the smart card or a passphrase question and answer) and how to handle scenarios such as lost cards.

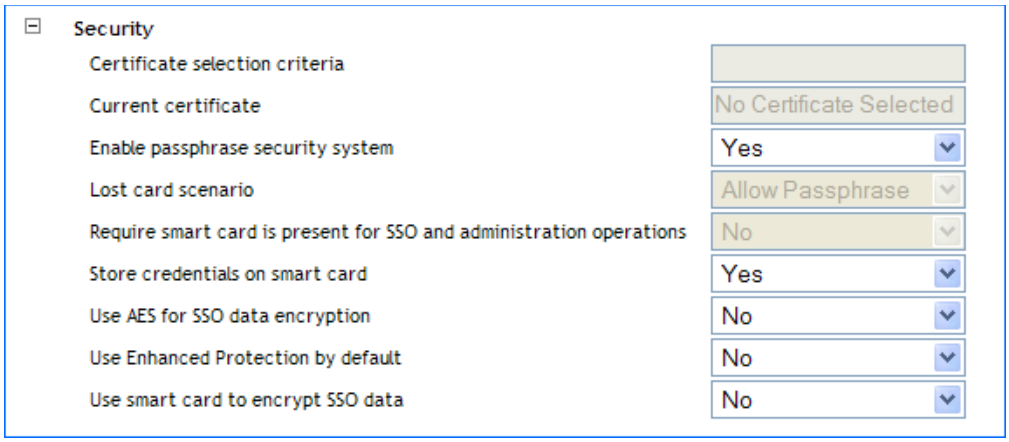
To configure the preferences, use the iManager in eDirectory environments, MMC plug-in for Active Directory environments, and SecureLogin Manager in LDAP v3-compliant directories such as Sun*, Oracle*, and IBM*.

1. Access the Administrative Management utility of Novell SecureLogin.

   For information on accessing the Administrative Management utility see, Section 1.2, "Starting the Administrative Management Utilities," on page 14 and, or, Section 1.3, "Accessing the Single Sign-On Plug-In Through iManager," on page 15.

2. Click *Preferences*. The Preferences Properties table is displayed.

3. In the *Setting Description* column, go to *Security* and select the appropriate preferences.



4 Click *Apply*.

5 Click *OK*.

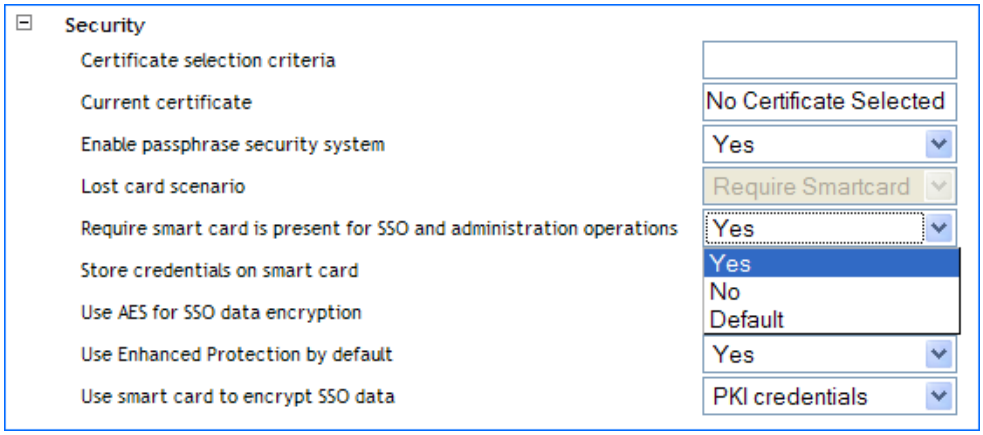
The following sections explain the various security preferences:

- Section 8.3.1, “Requiring a Smart Card for SSO and Administration Operations,” on page 87
- Section 8.3.2, “Storing User Credentials on Smart Card,” on page 89
- Section 8.3.3, “Using AES for SSO Data Encryption,” on page 90
- Section 8.3.4, “Using a Smart Card to Encrypt SSO Data,” on page 90
- Section 8.3.5, “Using PKI Encryption for the Datastore and Cache,” on page 92
- Section 8.3.6, “Certificate Selection Criteria,” on page 92

8.3.1 Requiring a Smart Card for SSO and Administration Operations

The *Require smart card is present for SSO and administrative operation* option determines if a user's smart card must be present before allowing a single sign-on session or administration function. This option also checks to see if a smart card has been removed after the start of a single sign-on session, which prevents the swapping of smart cards to copy a user’s credentials.

Figure 8-2 *Requiring a Smart Card for SSO and Administration Operations*



If the smart card is removed after the single sign-on session has started, and then reinserted, the card serial number is checked to validate that the card now being used is the same card used to initiate the single sign-on session.

For a new user if the credentials are stored on smart card with the certificate selection criteria is set as a friendly name, SecureLogin fails to launch. It displays a message indicating that, `The smart card does not contain any certificates that match the certificate selection criteria` is displayed. The user is forced to click *OK*, log out and log in again with either smart card or username.

This occurs because when a new user log in to Windows for the first time, Windows tries to set up the user's desktop. So, there is a delay in propagating the certificate from the smart card to the local store. As a result, the certificates are not available when SecureLogin launches and there is a delay in launching SecureLogin.

To launch SecureLogin successfully on the first attempt, change the value of FindCertificateRetryNumber and FindCertificateRetryInterval registries. These registries allow SecureLogin to retry finding certificate in the local store in a fast user switching environment or slow Windows startup.

The FindCertificateRetryNumber registry controls the number of times SecureLogin retries for the certificate. The FindCertifcateRetryInterval registry specifies the interval (milliseconds) to wait between each try.

1. Browse to `HKEY_LOCAL_MACHINE\SOFTWARE\Protocom\SecureLogin`
2. Set the DWORD value of FindCertificateRetryNumber to:

If set to	Then
4	Novell SecureLogin installer sets the value to four during installation. This means that SecureLogin tries four times with default interval of four seconds in between each try.
1	Set the value to 1 to stop the retry. This is also the minimum number of retries. This is also the default behavior if the setting is removed from the registry.
360	This is the maximum number of retries. SecureLogin reverts to this value if the value specified in the registry is greater than the maximum retries.

3. Set the DWORD value of FindCertificateRetryInterval to:

If set to	Then
5000	This is default value if setting is not specified in the registry. SecureLogin installer does not populate this setting during installation.
6000	The maximum value for the interval. SecureLogin reverts to this value if the value specified in the registry is greater than the maximum.

SecureLogin launches successfully on subsequent logins because Windows is not required to set up the user desktop and the certificate propagation happens before SecureLogin launches.

NOTE: If the *Lost card scenari*o option is set to *Allow passphrase,* then the *Require smart card is present for SSO and administration operations* option is dimmed and not available.

If *Lost card scenario* is set to *Require smart card*, then the *Require smart card is present for SSO and administration operations* option is available and defaults to *Yes*.

If you select *No*, the user's smart card is not required for single sign-on and administration operations.

If you select *Yes*, the user's smart is required for single sign-on and administration operations.

If the *Default* option is selected, this option is set to *No*. Alternatively, the user's credentials inherit the *Require smart card is present for SSO and administration operations* option set by the higher-level container.

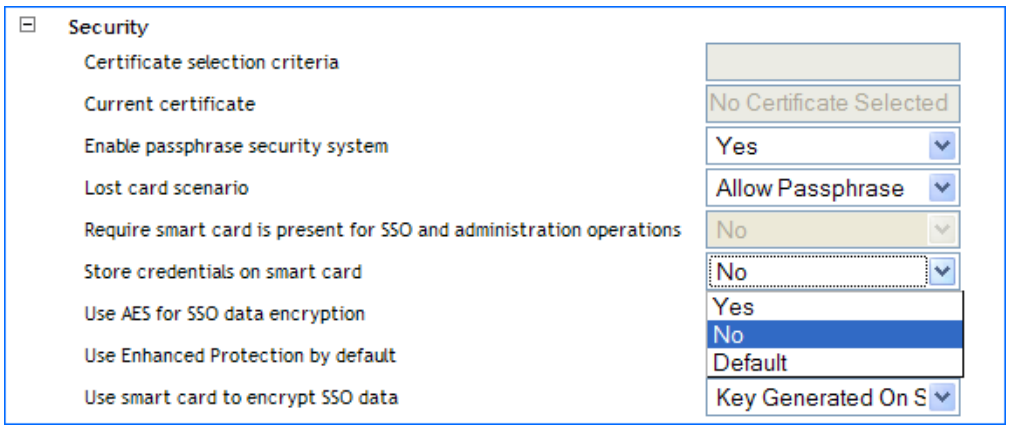
IMPORTANT: Any changes to the *Require smart card is present for SSO and administration operations* preferences requires SecureLogin to be closed and restarted before the changes takes effect.

You can manually disable inheritance of higher-level options by selecting the *Yes* option for *Stop walking here* (SecureLogin Administrative Management utility > *Preferences* > *General* options.)

8.3.2 Storing User Credentials on Smart Card

Use the *Store User credentials on smart card* option to select how user credentials are stored.

Figure 8-3 *Storing User Credentials on Smart Card*



If you select *No*, the user's credentials are stored in the user's local (off-line) cache.

If you select *Yes*, the user's single sign-on credentials, including usernames and passwords, are stored on the smart card in a secure PIN-protected container. Although credentials are stored on the smart card, other single sign-on data, including application definitions and preferences, are stored in the user's local cache on the hard drive.

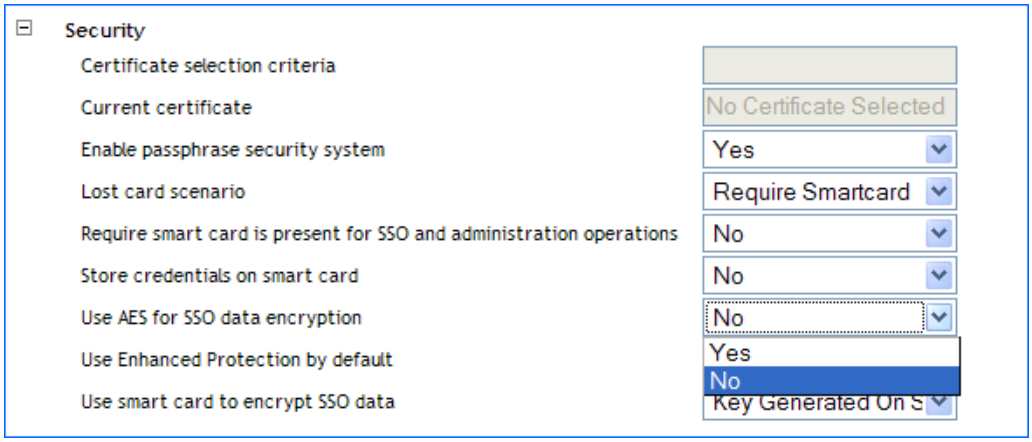
If the *Default* option is selected, the user's credentials are stored in the user's local (off-line) cache as with the *No* option. Alternatively, the user's credentials inherit the *Store credentials on smart card* option set by the higher-level container.

NOTE: You can manually disable inheritance of higher-level options by selecting the *Yes* option for *Stop walking here* (SecureLogin Administrative Management utility > *Preferences* > *General* options.)

8.3.3 Using AES for SSO Data Encryption

This option determines the level and standard of encryption used to encrypt single sign-on data stored on the smart card by allowing the use of AES instead of triple DES.

Figure 8-4 *Using AES for SSO Data Encryption*



If you select *No*, a 168-bit key used with triple DES (EDE) in Cipher-Block Chaining (CBC) mode is used to encrypt the user's single sign-on credentials.

NOTE: The input key for DES is 64 bits long and includes 8 parity bits. These 8 parity bits are not used during the encryption process, resulting in a DES encryption key length of 56 bits. Therefore, the key strength for Triple DES is actually 168 bits.

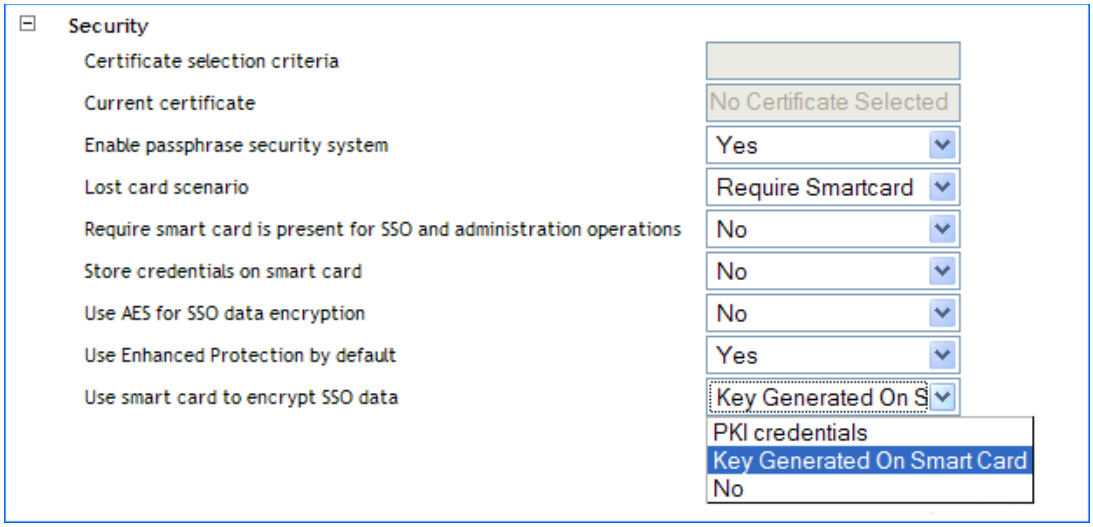
If you select *Yes*, then a 256-bit key used with AES (EDE) in CBC mode is used to encrypt the user's credentials.

If a previous version of SecureLogin has been implemented with passphrases enabled and if this option is set to *Yes*, users must answer with a passphrase before data can be decrypted and reencrypted by using AES.

8.3.4 Using a Smart Card to Encrypt SSO Data

SecureLogin 6.1 offers various encryption options. By default, SecureLogin encrypts data using either a user-defined passphrase key or a randomly generated key. The *Use smart card to encrypt SSO data* option can be used to determine whether PKI credentials or the self-generated key are stored on the smart card and then used to encrypt the user's single sign-on data.

Figure 8-5 *Using a Smart Card to Encrypt SSO Data*



If you select *No*, all other smart card options are dimmed and not available.

If you select *PKI credentials*, single sign-on data is encrypted by using the user's PKI credentials. Single sign-on data stored in the directory and in the offline cache (if enabled) is encrypted by using the public key from the selected certificate, and the private key (stored on a PIN protected smart card) is used for decryption.

If you select *Key generated on smart card* option, single sign-on data is encrypted by using a randomly generated symmetric key that is stored on the user's smart card. This key is used to encrypt and decrypt single sign-on data stored in the Directory and in the offline cache (if enabled).

NOTE: It is possible to inadvertently set these options to *Require smart card* under the following circumstances: First, you change the *Use smart card to encrypt SSO data* option to *PKI credentials*, then you change the *Lost card scenario* option to *Require Smartcard*, and finally change the *Require Smart Card is present for SSO and administration operations* option to *Yes*. If you do this, then both the *Lost card scenario* and R*equire smart card for SSO and administration operations* are set to *Require smart card*.

You should set these preferences in the following order:

1. Set the *Store credentials on smart card* to *No*.
2. Set the *Use smart card to encrypt SSO data* option to *PKI credentials*.
3. Click *Apply*.
4. Close and then reactivate SecureLogin. Check to see if the options are correctly set.

When a smart card is deployed with a user's PKI credentials, consider using key escrow, archiving, and backup through an enterprise card management system for the user's private key to be recovered in a lost card scenario. If no escrow is used, the *Enable passphrase security system* option should be set to *Yes* or *Hidden* to prevent the loss of the user's single sign-on credentials if a user loses a card.

8.3.5 Using PKI Encryption for the Datastore and Cache

If PKI credentials are used to encrypt single sign-on data and the passphrase security system is set to *No*, you should consider implementing a key archive for backup and recovery. If this system is not implemented and the passphrase security system is not enabled, users can never decrypt their single sign-on data if they lose a smart card, because the private key is stored on the smart card and is not recoverable.

Without private key recovery, if the user loses his or her smart card, the single sign-on administrator must clear the user's single sign-on data store and reset the back-end password before the user is able to use single sign-on again. This is a high security solution, but is more inconvenient to end users because they cannot have single sign-on access without the smart card.

For more information, refer Section 8.5.7, "Using a Card Management System," on page 98.

Choosing a Certificate

When a smart card is configured to use PKI credentials to encrypt single sign-on data, SecureLogin retrieves the serial number of the current certificate and locates the certificate in the certificate store as specified in the relevant SecureLogin preferences.

Figure 8-6 *Choosing a Certificate*

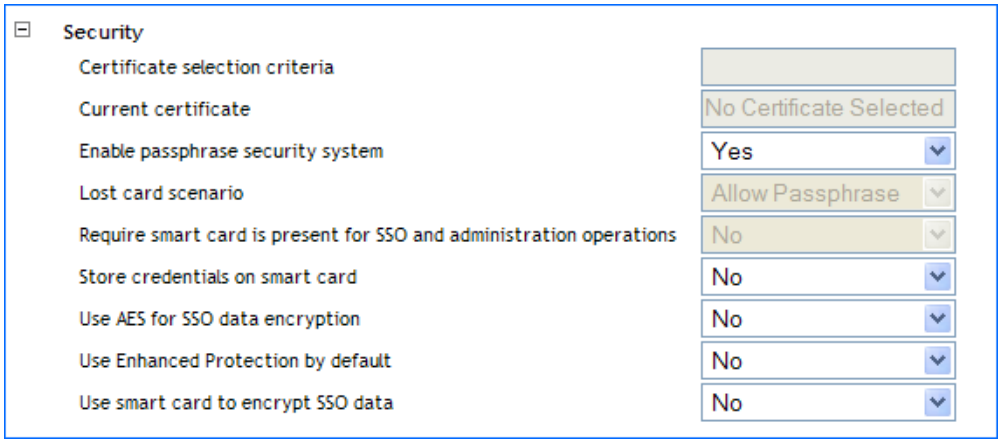
SecureLogin then loads the associated private key (which might cause a PIN prompt), and attempts to decrypt the user key with the private key.

In cases where the decryption fails or the certificate cannot be located but a smart card is present and a certificate that matches the selection criteria can be located, SecureLogin assumes that the recovered smart card is in use. SecureLogin the attempts to decrypt the user key with each key pair with the key pair stored on the card.

8.3.6 Certificate Selection Criteria

The *Certificate Selection Criteria* option allows you to select an encryption or authentication certificate to encrypt user's single sign-on information in the directory.

Figure 8-7 *Certificate Selection Criteria*



The certificate selection criteria determine which certificate to select if multiple certificates are in use (for example, if an enterprise has configured an Entrust* certificate for single sign-on encryption and a Microsoft certificate for login and or, authentication).

If only one certificate is used, the field is blank and the certificate is detected automatically and set to User Certificate. When entering certificate selection criteria, no special formatting is required and the search string is not case sensitive. Wildcards are not used and a search matches if the search text is a substring of the certificate subject field. SecureLogin attempts to match against the *Certificate Subject*, then the *Certificate Issuer* and finally the *Friendly Name* in that order.

For example if the subject is

```
CN=Writer,OU=Users,OU=Accounts,OU=APAC,DC=Novell,DC=Int
```

Then Writer is a valid search value, as are *Accounts*, *APAC*, and *Int*. The prefixes CN=, OU=, or DC= are not required.

Similarly, if the *Certificate Issuer* is

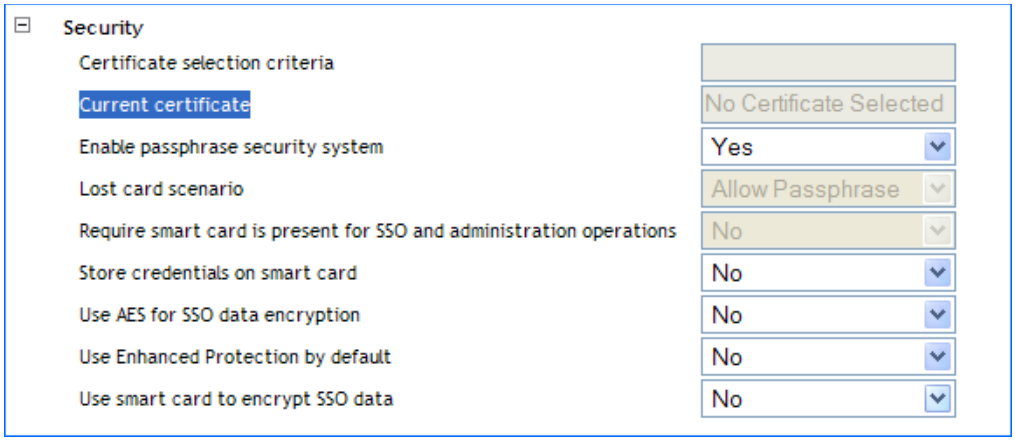
```
CN=IssuingCA1,OU=AD,DC=undiscovered,DC=com
```

Then *IssuingCA1* is a valid search value, as are *AD*, *undiscovered*, and *com*.

Current Certificate

This option displays the certificate that is currently being used by SecureLogin to encrypt a user's single sign-on data.

Figure 8-8 *Current Certificate*



8.4 Application Re-authentication with SLAA or NMAS

With SecureLogin, a user normally runs an application and SecureLogin seamlessly retrieves the user's application credentials. The credentials are authenticated in the background and the user is not prompted to enter a password. SecureLogin can also be configured to prompt the user (or a supervisor) for stronger authentication to all or specific applications. SecureLogin can be configured to request application re-authentication using SecureLogin's application definition `AAVerify` command.

The `AAVerify` command can enforce stronger application-based re-authentication such as biometrics, tokens, or smart cards when the native application cannot enforce strong verification. *AAVerify* works by requesting the preconfigured strong re-authentication method before SecureLogin will retrieve and enter the username and password for the application.

You can configure which applications require `AAVerify` (re-authentication) and which do not. The application itself is not changed and no additional modules are required on the application servers.

NOTE: SecureLogin 6.0 and above require SecureLogin Advanced Authentication 1.93.5 and above to utilize `AAVerify`.

8.4.1 Re-authenticating Individual Applications

SecureLogin 6.0 and later now allow you to set the re-authentication method for user's individual applications by using SecureLogin's Administrative Management utility *Application* > *Settings*. Individual applications can be re-authenticated against an advanced authenticating device, where SecureLogin is used in conjunction with SecureLogin Advanced Authentication or NMAS without running a dedicated application definition.

8.4.2 Scripting for One-Time Passwords

The SecureLogin application definition `GenerateOTP` command is enhanced to incorporate the one-time password soft token generation functionality that is embedded in ActivClient smart cards.

This one-time password functionality can only be used with ActivClient and smart cards that have been set up using a card management system to include a one-time password applet on the smart card.

Synchronous Mode

Synchronous authentication or time-plus-event authentication replaces static alphanumeric passwords with a pseudo-random code that is dynamically generated at configured time intervals, generally about 60 seconds. The code is based on a shared encryption key and the current time.

In Synchronous mode, the `GenerateOTP` command requires the administrator to pass a mode variable to the command.

Asynchronous mode

Asynchronous authentication or challenge and response authorization replaces static alphanumeric passwords with a pseudo-random code that is dynamically generated based on a shared encryption key, the current time, and a challenge/response combination. In asynchronous mode the challenge is passed to the `GenerateOTP` command as an argument.

8.5 Lost Card Scenarios

The *Lost Card Scenario* option determines how SecureLogin handles a user forgetting, losing or damaging a smart card. The *Lost Card scenario* option can only be used if, the *Enable passphrase security system* option has been enabled (the *Yes* or *Hidden* options).

If the lost smart card is being used to encrypt single sign-on data and a card is lost or stolen or damaged, and key escrow or recovery is not used, the user does not have access to single sign-on data unless *Enable passphrase security system* is set to *Yes* or *Hidden*.

If *Enable passphrase security system* is set to *Yes*, if the user has previously set a passphrase, and if *Lost card scenario* is set to *Allow Passphrase*, the user is prompted to answer with his or her passphrase before SecureLogin is available.

If *Enable passphrase security system* is set to *Hidden*, the user is not prompted for the answer and SecureLogin loads seamlessly.

8.5.1 Requiring a Smart Card

The *Require smart card* option prevents a user from starting single sign-on without his or her smart card. This option is for high security implementations where organizations want to tie the use of a user's single sign-on credentials to the user's smart card. This means that the user cannot access single sign-on with any other method; that is, they cannot use a username and password without the smart card.

IMPORTANT: If the *Require smart card* option is changed while the user is logged in, refreshing the cache using the *Advanced > Refresh Cache* option from the taskbar does not refresh the *Lost card scenario* option.

The user must log out and log in again (or restart SecureLogin) for the new option to take effect.

8.5.2 Allowing a Passphrase

The *Allow passphrase* option must be used in conjunction with the *Enable passphrase security system* option. It allows the user to start SecureLogin by using a passphrase if the smart card is not available. The passphrase security system must be set to *Yes* or *Hidden* for this setting to apply.

The *Hidden* option replaces a user-generated passphrase with a system-generated passphrase, effectively removing the need for the user to remember the passphrase answer.

IMPORTANT: For the user to decrypt data using a passphrase, the passphrase must already be set. You cannot simply toggle the *Enable passphrase security system setting* to on the day the user forgets a smart card unless the user has previously set a passphrase (or had it randomly generated by using the *Hidden* option).

The *Default* option allows the user to start SecureLogin by using a passphrase if the smart card is not available through the *Allow Passphrase* option. Alternatively, this option inherits the *Lost Card scenario* preference set by the higher-level container.

NOTE: You can manually disable inheritance of higher-level options by selecting the *Yes* option for *Stop walking here* (SecureLogin Administrative Management utility > *Preferences* > *General* options.)

8.5.3 Passphrases for Temporary Access

There are a number of options available that permit access if a user loses or forgets his or her smart card. For example, If a user loses or forgets his or her smart card and the *Lost card scenario* option is set to *Require smart card*, you can grant temporary access to systems by resetting the user's password. The user is then required to log in and enter the passphrase. This option is possible only if the *Enable passphrase security system* is turned on.

However, the user should not expect easy or automatic access to the system. Users should understand that, a strong and secure solution has been implemented and that they have the responsibility of looking after their own smart cards.

8.5.4 Restoring a Smart Card Using Card Management System

If an enterprise opts to deploy corporate smart cards without a suitable card management system (CMS) user key escrow, archiving, and backup system combined, you can still create a very high level of security by setting *Enable passphrase security system* to *No* and selecting the *Use smart card to encrypt SSO data* options of *PKI credentials* or *Key generated on smart card* options. However, in the event of a lost or damaged smart card the user can never decrypt the single sign-on data because the key stored on the smart card is not recoverable.

If this is the case, you need to delete the user's existing single sign-on configuration data store from the *Advanced Setting* > *Datastore* tab.

Deleting the user's single sign-on datastore permanently deletes all the user's corporate applications, credentials, options, and user policies.

You must then reset the user's corporate passwords and issue a new smart card (with a new key pair) before the user can log in and reconfigure the single sign-on applications using SecureLogin again.

The user must manually enter all application credentials into SecureLogin the first time he or she logs in after the data was cleared from the directory.

Enterprises should consider implementing key escrow, archiving, or backup through a suitable CMS to allow a user's encryption key to be recovered in the event of a lost or damaged smart card.

The use of a CMS is crucial if an enterprise opts to deploy corporate smart cards with a very high level of security by disabling the *Enable passphrase security system option* combined with using the *Store credentials on smart card* set to *Yes* and the *Use smart card to encrypt SSO* data options of *PKI credentials* or *Key generated on smart card options*.

In the event of a lost or damaged smart card, the user can never decrypt their single sign-on data because the key stored on the smart card is not recoverable.

IMPORTANT: It is recommended that you extensively test the CMS and smart card restoration techniques before selecting the high security options described above that tie single sign-on to the user's smart card.

8.5.5 PKI Credentials

If the *Use smart card to encrypt SSO* data option is set to use *PKI credentials* to encrypt a user's single sign-on data and *Enable passphrase security system* is set to *No*, in the event of a lost or damaged smart card the user can never decrypt the single sign-on data because the key stored on the smart card is the only key that can be used for decryption and is not recoverable unless key archiving and recovery are implemented.

If a CMS-based key archive is used, then the encryption key needs to be recovered to the new smart card, the single sign-on data unencrypted, and an administrator needs to chose a new certificate to encrypt the user’s data.

If you are using the enterprise CMS-based recovery system, you must issue the user a replacement smart card based on a CMS backup of the user's original key.

8.5.6 Key Generated on Smart Card

Similarly, if the *Use smart card to encrypt SSO* data option is set to use *Key generated on smart card* to encrypt a user's single sign-on data, then in the event of a lost or damaged smart card the user can never decrypt the single sign-on data because the key stored on the smart card and is not recoverable.

You should consider setting the *Enable passphrase security system* option to *Yes* when the *Key generated on smart card* option is used to provide an alternative mechanism for decrypting single sign-on data if the smart card is lost/stolen/damaged.

Using the enterprise CMS-based recovery system, the administrator must issue the user a replacement smart card based on a CMS backup of the user's original key. The replacement card includes the recovered private key and a new key pair so data can be decrypted using the old key and re-encrypted using the new key.

8.5.7 Using a Card Management System

Enterprise server or Web-based card management system software enables corporations to implement and easily manage smart card identity management, provisioning, authentication devices, and policy enforcement across geographically dispersed locations.

These systems provide a complete and flexible solution to manage the issuance, administration, and configuration required for the successful and seamless smart card integration with SecureLogin 6.0 and later and Smart Card Password Login.

Enabling Applications and Web Sites

9

This section has information on the following:

9.1 Enabling Applications and Web Sites for Single Sign-On

Novell® SecureLogin has the following features:

- Predefined applications for single sign-on to access a wide range of commercially available applications.
- The ability to detect applications for which a predefined application exists. For example, if SecureLogin detects a SAP login dialog box, then SecureLogin prompts the user with an option to allow SecureLogin to automatically enable the application for single sign-on.

  NOTE: Predefined applications for commonly used applications are provided with the SecureLogin application, and with each new version, more are developed and made available to the Novell customers.

- Wizards and application definitions to facilitate single sign-on to almost any new or proprietary application if a predefined application is not available. This helps you or Novell Support to build an application definition for almost any proprietary application or upgrade.
- Support for single sign-on-enabling of most standard terminal emulator applications.

- Additional single sign-on tools, such as the Window Finder and LoginWatch, which help you enable even the most difficult applications for single sign-on.

  NOTE: You can enable terminal emulators for single sign-on by using the Terminal Launcher tool.

- It stores the login information requirements for applications including:

Table 9-1 *Login Information Stored by Novell SecureLogin*

Credentials, including but not limited to:	◆ Username ◆ UserID ◆ Login ID ◆ Password ◆ PINs ◆ Domain ◆ Database names ◆ Server IP address
Responses to dialog boxes, messages and windows events, for example:	◆ Login ◆ Incorrect credentials ◆ Password expiry and reset ◆ Error messages, including non-compliance to password rules ◆ Account locked ◆ Database locked

Before SecureLogin can enable an application for single sign-on for a particular user, it must learn the user's application credentials so that SecureLogin can encrypt and store them for future logins, unless it is used in conjunction with Identity Management solutions such as Novell Identity Manager.

When a user starts an application for the first time after the application was enabled for single sign-on, SecureLogin prompts the user for application credentials, and then encrypts and stores them in the directory against the user object. The credentials are passed automatically to the application for subsequent logins.

Automated single sign-on is achieved by using proprietary application definitions. Application definitions are managed in directory environments through SecureLogin management utilities, including the Administrative Management utility, iManager plug-in, and Active Directory MMC plug-in. Locally and in standalone deployments, application definitions are managed in the Personal Management utility or distributed by using the advanced offline signed and encrypted method.

Applications that are enabled for single sign-on are created, modified, and deleted in the *Applications* pane. You can also create application definitions with SecureLogin wizards. There are a wide range of options in SecureLogin to enable applications. Regardless of the origin of the application definition, when an application is enabled for single sign-on, it is added and maintained in the *Applications Properties* Table.

9.2 Using the Add Application Wizard to Enable a Windows Application

The Add Application Wizard helps you build application definitions and enable Windows applications for single sign-on.

NOTE: The *Add Application* Wizard and the Administrative Management utility cannot be active simultaneously. Exit the Administrative Management utility before using the *Add Application* Wizard.

To add an application through the Wizard:

1. Start the required application to display the login.
2. In the notification area, right-click the Novell SecureLogin icon, then click *Add Application*. The Welcome to SecureLogin page is displayed.
3. Click *Next*. The Single sign-on enable an application page is displayed.
4. Select the appropriate option, then click *Next*. The Single sign-on enable a windows application login or message box page is displayed.
5. Specify your credentials such as user name, password and any other required information in the login dialog box.
6. Click and drag the hand icon onto the application's login title bar. The Select window function page is displayed.
7. In the drop-down list, click the appropriate option.
8. Click *Next*.The Select the appropriate fields and enter credentials page is displayed.
9. Click and drag each to the relevant box and release the mouse button to confirm selection.

   The check box to the left of the description changes to blue when a box or button is selected.
10. Click the *OK* button to the left of the and drag across to *OK* in the application's login window.
11. Click *Next*. The Name the Application Definition page is displayed.
12. Specify a name for your application definition or select one of the suggestions.

    The suggested names provided are based on the type of window function that the Wizard detected in the earlier steps, such as Login, or Change Password.
13. Click *Finish*.

    The Wizard closes and the application definition is created.
14. Close your application login window without logging on. SecureLogin enters your credentials and logs on to the application.

    The new application definition is now available to customize in the Applications pane of the Personal Management utility or Administrative Management utility.

9.3 Enabling Java Applications

Novell SecureLogin support single sign-on access to Java applets and applications implementing AWT and SWING GUI components, as well as JavaScript*. Both Java and JavaScript are included in the functionality labeled throughout the SecureLogin user interface. When a login dialog box is recognized by SecureLogin, a confirmation message appears.

9.3.1 Prerequisites

- Install a Sun Runtime Environment Version 1.4 or later and Oracle JInitiator 1.3.1 or later.

  NOTE: Microsoft Java Virtual Machine is not supported.

- Select the option for enabling applications during SecureLogin installation.
- Ensure that in the *Preference Properties* table, the value for *Add application prompts for applications* is set to *Yes*.
- Ensure that in the Preference Properties table, the value for *Allow single sign-on to Java applications* is set to *Yes*.

9.4 Using a Predefined Application to Enable a Web Application

The following example demonstrates enabling single sign-on for a Yahoo e-mail account. SecureLogin provides a predefined application for Yahoo mail.

NOTE: This procedure assumes that you already have a Yahoo e-mail account.

1. Start your web browser and go to *http://mail.yahoo.com*. The Enter your credentials dialog box is displayed.
2. Provide your username and password.
3. Click *OK*.

   SecureLogin saves your credentials and uses them to log in to your Yahoo mail account. Your Yahoo account displays with your credentials securely saved.
4. To test whether the single sign-on has been successful, sign out of Yahoo mail. A confirmation message appears. Click *OK*.
5. SecureLogin enters your credentials and logs you back to your Yahoo mail account.

   If your login is successful, your application is defined correctly.

   If your login is not successful, delete the application definition and repeat the above steps.

   You might also need to review the application definition for completeness of event responses and errors.

9.5 Using the Web Wizard to Enable a Web Site

1. Start your Web browser and navigate to the Web site containing the login fields.
2. Enter your login details. The Web login detected dialog box is displayed.
3. Click one of the following options:
   - **Yes:** To single sign-on-enable the Web site.
   - **Later:** To stop the enabling process for this session. You will be prompted to enable the Web site the next time you log in.
   - **Never:** To stop the enabling process for this Web site and never receive future prompts.
   - **Options:** To customize the description for this application.

SecureLogin captures your login details and adds them to your Web application definitions.

4 To view the application definition created above, start the Personal Management utility.

5 Click *Applications*. The application definitions are listed in the Applications pane.

6 In the navigation tree, under *Web*, double-click the application created by the Web Wizard. The *Details* tab lists the application definition.

7 Click the *Definition* tab.

NOTE: The Definition tab lists the application definition. The Definition tab allows you to customize site and credential details. Also available on this tab is an *Advanced* button which provides more functionality for these application definitions.

8 (Optional) Customize Site Properties by selecting the following:

If	Then
You want to automatically login to the Web site each time you go to it.	Select the *SecureLogin proceeds with login after entering your credentials* check box.
You have more than one login for a Web Wizard application.	Select the *Supply credentials* check box and click the appropriate credentials in the drop-down list.
You want to view the descriptions and associated site details.	Click *Advanced*.
You want to convert the details on the *Definition* tab to a SecureLogin application definition which displays as script.	Click *Convert To Script*. If you select *Convert To Script*, the action cannot be reversed. To enable an application, you must delete the existing Web Wizard application definition and repeat the process of enabling the Web site for single sign-on.

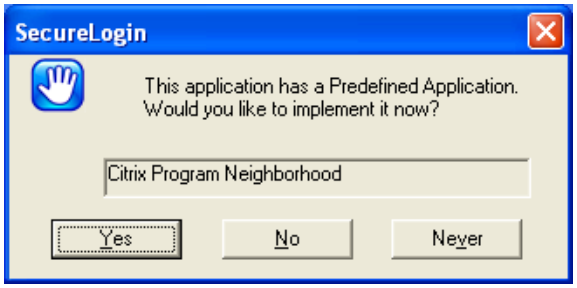
9.6 Using Predefined Application Definition to Enable Citrix Program Neighborhood

The following example demonstrates enabling Citrix Program Neighborhood for single sign-on by using a Novell SecureLogin predefined application definition.

NOTE: This procedure assumes that you have an existing Citrix Neighborhood account.

1 On the Windows *Start* menu, click *Programs* > *Citrix* > *Citrix Access Client* > *Program Neighborhood*. The CITRIX-PS4 dialog box is displayed.

2 Specify your username, password, and domain, then click *OK*. The SecureLogin dialog box displayed.



3 Click *Yes* to enable Citrix Program Neighborhood for single sign-on. The Enter your credentials dialog box is displayed.

4 Specify your username, password, and domain, then click *OK*.

SecureLogin saves your credentials and uses them to log in to your single sign-on enabled Citrix Program Neighborhood.

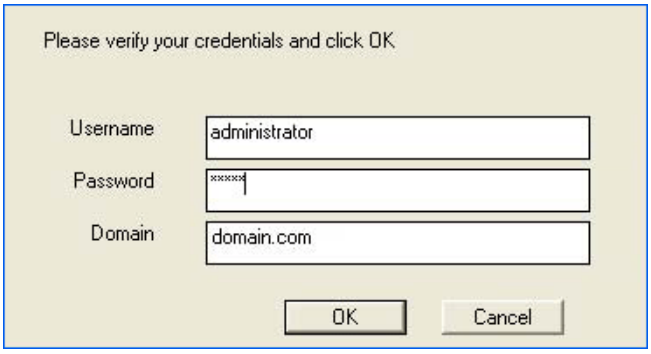
4a If you have specified incorrect credentials during the single sign-on verification process, the following message is displayed.

Another message indicating that a predefined application exists for the Citrix Program Neighborhood is displayed.

4b Click *Yes* on the SecureLogin dialog box to proceed with the re-verification process. The Enter you credentials dialog box is displayed again.

The Citrix Program Neighborhood dialog is automatically cleared when you click *Yes* on the SecureLogin dialog box.

If you do not click *Yes* on the SecureLogin dialog box, the original single sign-on enabling process is resumed with the incorrect credentials.



4c Specify your username, password, and domain, then click *OK*.

To test verify if your Citrix Program Neighborhood account is successfully enabled for single sign-on, log out of Citrix Program Neighborhood and login again.

9.7 Using the Add Application Wizard to Enable a Web Site

The Add Application Wizard helps you enables Web sites for single sign-on.

The *Add Application* Wizard and the Administrative Management utility cannot be active simultaneously. Exit the Administrative Management utility before using the wizard.

To enable a Web site by using the *Add Application* Wizard:

1. Go to the Web site's login page.
2. In the notification area, right-click the Novell SecureLogin icon, then click *Add Application*. The Welcome to SecureLogin page is displayed.
3. Click *Next*. The Single sign-on enable an application page is displayed.
4. Select the appropriate option, then click *Next*.The Single sign-on enable a web/Internet application page is displayed.
5. Copy and paste the Web site's URL into the URL field. Click *Finish*.

   The Web site is now enabled for single sign-on and you will be automatically logged in to the Web site the next time you visit.

9.8 Enabling Terminal Emulator Applications

You can configure terminal emulators for single sign-on in the application definition editor in the Administrative Management utility, in the Personal Management utility, and the Terminal Launcher tool.

To enable a terminal emulator for single sign-on, you must run `tlaunch.exe`, which you configure in Terminal Launcher, and link to the configuration in an application definition.

Terminal Launcher helps you configure terminal emulator applications for single sign-on. The following sections document these procedures to do the following through an example application:

- Creating and save a terminal emulator session file.
- Building a terminal emulator application definition.
- Running Terminal Launcher.
- Creating a terminal emulator desktop shortcut.
- Setting Terminal Launcher command line parameters.

Although these procedures apply to most terminal emulators, the application definition and other configuration information might differ for each emulator application. Contact Support for help.

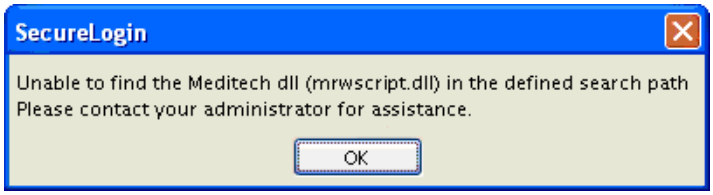
9.8.1 Support for the MEDITECH Predefined Application

Novell SecureLogin 6.1 SP1 supports MEDITECH* 3.*x* and 4.*x*. It is dependant on the mandatory presence of the MEDITECH `mrwscript.dll`. The `.dll` file is provided by MEDITECH and must be installed during the installation of the MEDITECH application on the workstation.

NOTE: If you are an existing customer of Meditech, you can obtain the `mrwscript.dll` as part of your MEDITECH support agreement.

During the installation of the predefined MEDITECH application, SecureLogin detects the presence of the file and immediately warns, if the file cannot be located.

Figure 9-1 *MEDITECH Warning Message*



9.9 Creating and Saving a Terminal Emulator Session File

Prior to enabling any terminal emulator for single sign-on, you must identify or create a session file that includes all the required settings for the server connection and any other parameters required for deployment to users. Terminal Launcher is configured to run this session file when launching the emulator. Any modifications to the session must be saved to this file. The session file can be saved locally or on the server. Typically, the session file already exists and you just need to configure Terminal Launcher to point to the relevant file.

If you need to create a session file:

1. Start the terminal emulator application.
2. Connect to the required host.
3. Change the terminal emulator settings as required.
4. Save the session. The default directory is usually the application's installation directory.
5. On the *Connection* menu, click *Disconnect*.The session file remains loaded, but you have disconnected from the host.
6. On the *File* menu, click *Save* [session name] to save changes to the session file.
7. Exit the terminal emulator application.

9.10 Building a Terminal Emulator Application Definition

In the following procedure, you build a terminal emulator application definition on the local workstation for the example application Eicon* Aviva*.

1. Open the Personal Management utility of SecureLogin by double-clicking , or by selecting *Start* > *Programs* > *Novell SecureLogin* > *Novell SecureLogin*.
2. Select *File* > *New* > *Application*. The New Application dialog box is displayed.
3. Select *New Application Definition*.
4. In the *Type* drop-down list, click *Terminal Launcher*.
5. In the *Name* field, specify a name for the application definition (in this example, Eicon Aviva), then click *OK*. The new application definition is added to the Applications pane.
6. Double-click the new application definition. The *Details* tab is displayed.

7. Click the *Definition* tab. The application definition editor is displayed.
8. Delete the default text displayed in the text box: `# place your application definition here`
9. In this example for Eicon Aviva, type the following in the text box:

```
WaitForText "WELCOME TO THE EICON TECHNOLOGY DATA CENTER "
Type @E
WaitForText "ENTER USERID -"
Type $Username
Type @E
WaitForText "Password  ===>"
Type $Password
Type @E
WaitForText " Welcome to Eicon Technology"
WaitForText "***"
Delay 1000
Type @E
```

   You must type the screen syntax accurately in the application definition editor; otherwise it will fail to operate. Wherever possible, cut and paste the text directly from the emulator screen into the editor.
10. Click the *Details* tab.
11. Ensure that the *Enabled* check box is selected.
12. Click *OK*.

9.11 Running a Terminal Launcher

Terminal applications require Terminal Launcher to execute for single sign-on. After you create the application definition in the Management utility, you must configure it to start Terminal Launcher. A shortcut is created to enable the user to run Terminal Launcher and the terminal emulator from the desktop with automated single sign-on to the application or server.

1. Select *Start* > *Programs* > *Novell SecureLogin* > *Terminal Launcher*. The Terminal Launcher dialog box is displayed.
2. In the *Available applications list*, click the required application definition (in this example, Eicon Aviva).
3. Click *Add* to move the selected application to the *Login to* list.
4. Click *Edit Available Emulators*. The Available Emulators dialog box is displayed.
5. In the *Available Emulators* list, click *Eicon Aviva*.
6. Click *Edit*. The HLLAPI Emulator Configuration dialog box is displayed.
7. In the *Emulator Path* field, specify the emulator executable's location.
8. In the *Home Directory* field, specify the emulator's home directory.
9. In the *HLLAPI DLL* field, specify the file name and path.
10. In the *Session Files* field, select and delete the current session files.
11. Click *Add*. The Emulator Session File dialog box is displayed.
12. Browse and select the configured session file.
13. Click *OK* to close the Emulator Session File dialog box.
14. Click *OK* to close the HLLAPI Emulator Configuration dialog box.

15 Click *Done* to close the Available Emulators dialog box.

16 In the Terminal Launcher dialog box, ensure that Eicon Aviva is selected in the Emulator drop-down list.

17 Under *Options*, select the *Save Settings On Exit* check box.

18 Click *Close*.

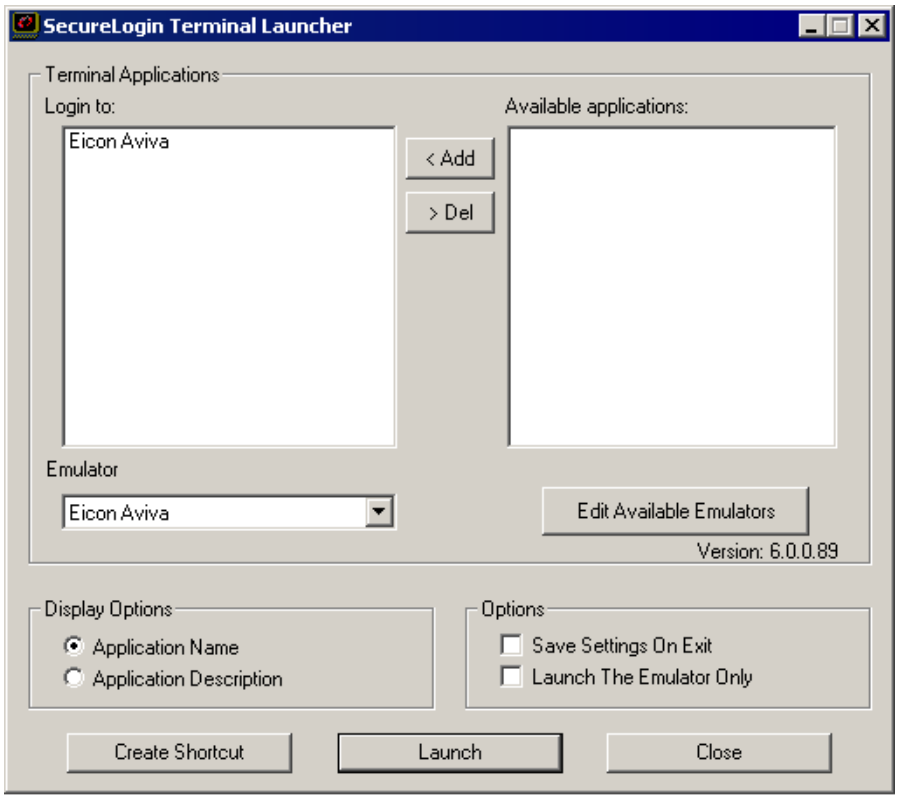
You can choose to start emulator applications in Terminal Launcher; however, users might not have access to Terminal Launcher. To simplify login for users, a desktop shortcut is created.

To successfully automate single sign-on, Terminal Launcher must start before the terminal emulator application, so the desktop shortcut includes the command to run Terminal Launcher first and then the emulator application.

NOTE: Record the exact name given to the terminal emulator in the Terminal Launcher dialog box, because it is referred to in the desktop shortcut.

9.12 Creating a Terminal Emulator Desktop Shortcut

1 Select *Start* > *Programs* > *Novell SecureLogin* > *Terminal Launcher*. The Terminal Launcher dialog box is displayed.



2 Click *Create Shortcut*. The Terminal Launcher Shortcut Options dialog box is displayed.

3 Select *Location* > *Desktop*.

4 Select the appropriate options from *Options*.

NOTE: *Quiet mode* and *Suppress errors* are the default options.

5 In the *Command Line* field, ensure that the following parameters are included (in this example, `/auto /e"Eicon Aviva" /pEicon Aviva /q /s`):

Parameter	Description
`/auto`	Indicates to Terminal Launcher that the following is a parameter requesting the execution of a terminal emulator application that is configured for single sign-on. This parameter is mandatory.
`/e[application name]`	Initiates the execution of the terminal emulator.
`/p[Terminal Launcher config name]`	Initiates execution of the application created in Terminal Launcher.
`/q`	Quiet mode (no Cancel dialog box).
`/s`	Suppress errors.

6 Add additional parameters as required.

7 Click *Create*.

The shortcut is created on the desktop and you can deploy it to users in the preferred mode for your organization.

8 Click *Close* to close the Terminal Launcher dialog box.

9 Double-click the short cut.

The terminal emulator application is executed with Terminal Launcher and the Enter your credentials dialog box is displayed.

10 In the *Enter login credentials* fields, specify your username and password.

11 Click *OK*.

SecureLogin stores the login credentials and uses them to log on to the application or a server. Subsequently, double-clicking the desktop shortcut logs the user directly on to the application or a server.

9.13 Setting Terminal Launcher Command Line Parameters

To run the required terminal emulator, Terminal Launcher command line parameters are included in the desktop shortcut command. For more information, see Section 9.12, “Creating a Terminal Emulator Desktop Shortcut,” on page 109.

The following table lists the parameters (also referred to as switches) you can set in conjunction with commands.

Table 9-2 *Terminal Launcher Command Line Parameters*

Parameter	Description
`/auto`	Indicates to Terminal Launcher that the following is a parameter requesting the execution of a terminal emulator application that is configured for single sign-on. For example: `C:\<....>\TLaunch.exe /auto /pApplication1` **NOTE:** This parameter is mandatory.
`/p[platform/ application/ Application Definition name]`	Initiates the execution of the terminal emulator as listed in the *Terminal Launcher Login to* field. To run multiple applications from the same command, add `/p[TL application/Application Definition name]` You can run up to fifteen applications simultaneously from the shortcut command line. For example: `C:\<....>\TLaunch.exe /auto /eEicon Aviva /pApplication1 /pApplication2` **NOTE:** You must type the emulator name exactly as it appears in the *Terminal Launcher Available Emulators* drop-down list.
`/b`	Specifies the background authentication mode.
`/e[emulator name]`	The parameter `/e[Terminal Launcher config name]` initiates the execution of the terminal emulator as listed in the *Terminal Launcher Available Emulators* drop-down list. **NOTE:** You must type the emulator name exactly as it appears in the *Terminal Launcher Available Emulators* drop-down list.
`/h[hllapi short name]`	Commands `TLaunch.exe` to connect to the specified HLLAPI session.
`/k[executable name]`	Quits (kills) the specified executable prior to launching the terminal emulator.
`/m`	Enables multiple concurrent connections to specified sessions. This parameter is required for background authentication.
`/n`	Starts the selected terminal emulator without executing a SecureLogin application definition. For example: `C:\<....>\TLaunch.exe /auto /n` **NOTE:** This parameter does not function with VBA emulators.
`/n[number 1-15]`	Starts the specified number of terminal emulator sessions without executing SecureLogin application definition. For example: `C:\<....>\TLaunch.exe /auto /n3` **NOTE:** This parameter does not function with VBA emulators.

Parameter	Description
/q	*Quiet Mode* (no Cancel dialog box). For example: C:\<....>\TLaunch.exe /auto /q
/s	Suppress errors.
/t	Unlimited timeout during connection. For example:C:\<....>\TLaunch.exe /auto /eEicon Aviva /pBackground /b /t /m /hA /s /q

9.14 Applications Excluded for Single Sign-On

Although Novell SecureLogin facilitates you to enable single sign-on for Windows, Web, and Java applications; some applications cannot be enabled for single sign-on. The applications that cannot be enabled include certain installers, Novell SecureLogin and Windows system files. Enabling these applications might affect your computer's performance or create a security risk.

These applications are hard-coded and are excluded from single sign-on.

Table 9-3 *Applications excluded from Single Sign-On*

setup.exe	Nwadmn95.exe	acsagent.exe
_isdel.exe	loginw95.exe	adamconfig.exe
msiexec.exe	NWTray.exe	rdbgwiz.exe
MSDEV.exe	loginw32.exe	ProtocomSysTray.exe
devenv.exe	scrnlock.scr	ac.aac.run.exe
SLBroker.EXE	MMC.EXE	Nwadmnnt.exe
tlaunch.exe	slwinsso.exe	ConsoleOne.exe
SLProto.exe	SLManager.exe	SLLauncher.exe
nswebsso.exe	sllock.scr	Nwadmn32.exe

9.14.1 Modifying The List

Although the applications disabled for single sign-on are hardcoded, you can modify the behavior by creating a text file at the <*Novell SecureLogin Install path*> For example, at C:\Program Files\Novell\SecureLogin\ and name it exclude.ini.

NOTE: Despite its extension, the exclude.ini file is not in an .ini file format.

You can open this file in any text editor and make the changes. You can extend or modify the list.

You can modify the file in the following ways:

- "Extending The List of Applications" on page 113

- “Including Applications For Single Sign-On” on page 113
- “Disabling The Default Behavior” on page 113

Extending The List of Applications

If you want to disable more applications apart from the hardcoded applications, add the names of the application to the exclude.ini file. For example, you can add `grpwise.exe` to the exclude.ini file. With this, GroupWise is also disabled for single sign-on.

NOTE: If you add an existing application to the list of applications in the exclude.ini file, it does not impact the original list. For example, if you add `SLProto.exe` to the exclude.ini file, it does not impact the function although it is listed twice.

Including Applications For Single Sign-On

If you want to enable only a set of applications for single sign-on, use `Include` keyword in exclude.ini file

In the exclude.ini file add the `Include` keyword to enable an executable for single sign-on. By including the Include keyword, the list is converted to an include list.

For example, when you add

```
Include

Trillian.exe
```

Trillian application is enabled for single sign-on. The next time you log in, you are prompted to enable single sign-on.

Disabling The Default Behavior

If you want to define a custom list for disabling the applications for single sign-on, include the `NoDefault` keyword. When you include the NoDefault keyword, the hardcoded applications are overrided.

For example, if you modify the list as:

```
NoDefault

NMCL32.exe
```

the hardcoded applications that are disabled for single sign-on is not read by Novell SecureLogin. Instead, the executables listed with the `NoDefault` keyword in the exclude.ini file are considered and all the applications listed in the file are disabled for single sign-on.

Reauthenticating Applications

10

SecureLogin Advanced Authentication (SLAA) allows you to reauthenticate an application against an AA device where SecureLogin is used in conjunction with SLAA or the Novell® NMAS™ infrastructure.

Use the procedure in this section if you have SLAA or NMAS in place against an application:

NOTE: For environments that use the Novell NMAS infrastructure, you can add the NMAS method in the Reauthentication Method value by providing a free text string from Novell.

---

1. For information on accessing the Administrative Management utility see, Section 1.2, "Starting the Administrative Management Utilities," on page 14 and, or, Section 1.3, "Accessing the Single Sign-On Plug-In Through iManager," on page 15.
2. Click *Applications*. The Application pane is displayed.
3. Double-click the application that you want to use for reauthentication.
4. Click the *Settings* tab. The Settings Properties table is displayed.
5. Set the value for *Prompt for device reauthentication for this application* to *Yes*.
6. From the *Reauthentication Method* drop-down list, select the device that you will use for reauthentication. Click *Any* if you want the user to choose from any of the available methods.

   NOTE: This option is not available through the iManager SSO plug-in

   ---

Adding Multiple Logins

11

Novell SecureLogin allows you to enable multiple logins for single sign-on to the same application. Before enabling your additional logins for single sign-on, make a list, including usernames and passwords, with a name to uniquely identify the login. The following is an example list:

Table 11-1 *List of Additional Logins*

Name	User Name	Password
Administrator	admin	123456
Support	help	abcdef
User	test1	xyz123

When the list is completed, use it to provide information as you complete the following procedure:

1. Enable the first account for single sign-on.

   For more information, see Section 9.5, "Using the Web Wizard to Enable a Web Site," on page 102.
2. In the notification area, right-click the Novell SecureLogin icon, then select *New Login*. The Add New Login Wizard Welcome page is displayed.
3. Select the enabled application.
4. Click *Next*. The Add New Login page is displayed.
5. In the *Description* field, specify a descriptive name for the login.
6. Click *Finish*. The Enter your credentials dialog box is displayed.
7. In the *Username* field, specify your user name.
8. In the *Password* field, specify your password.
9. Specify any additional variables as required, then click *OK*.
10. Repeat Step 1 on page 117 through Step 9 on page 117 to add any additional logins as required.

    When you have created all logins with the *Add New Login* Wizard, you can view them and manage them in the Personal Management utility.
11. In the notification area, double-click to open the Personal Management utility.
12. Click *My Logins*. The My Logins pane is displayed.
13. Verify that the additional login is added to the *My Logins* pane, then click *OK* to close the Personal Management utility.
14. Log in to the application with multiple SecureLogin accounts. Start the application.

    The [application] login selection dialog box is displayed.
15. Select the required login credential set, then click *OK*.

    SecureLogin enters the credentials, and you are automatically logged on to the application.

Managing Application Definitions 12

An application definition is a list of instructions that Novell SecureLogin follows to perform various tasks in various windows and dialog boxes.

For example, for a Windows application (`*.exe`), an application definition is written for each executable file that you want Novell SecureLogin to act upon. In that application definition you are can assign different instructions to each dialog box or screen that an executable file or application might produce. You can create actions for only the login panel, only selected windows, or every window that is produced by the executable file, such as account locked, invalid username, invalid password, back-end database is down, password expiry, and so on.

For detailed information on application definitions see, the *Novell SecureLogin 6.1 SP1 Application Definition Guide*.

This section provides information on the following:

12.1 Adding Support for Password Changes

Depending on your organization's policies regarding password expiration, users might be required to change their passwords on a regular basis. Each time user password is changed for an application that is enabled for single sign-on, SecureLogin must update the password data. To ensure that user password changes are updated in SecureLogin, it is important to configure SecureLogin to respond to the Change Password dialog box.

Using the Add Application Wizard, you can configure SecureLogin to automatically generate a new password (according to password policy, if required) whenever the Change Password dialog box is displayed. A randomly generated password is safer than user-defined, reusable passwords.

IMPORTANT: The Change Password dialog box must be displayed for the Add Application Wizard to identify it.

---

1 Display the Change Password dialog box.

2 On the notification area, right-click the Novell SecureLogin icon, then click *Add Application*. The Welcome to SecureLogin page is displayed.

3 Click *Next*. The Single sign-on enable an application page is displayed.

4 Click *Next*. The Single sign-on enable a Windows application login or message box is displayed.

5 Click and drag the onto the application's login title bar. The Select window function page is displayed.

6 In the drop-down list, select *Change Password Window*.

7 Click *Next*. The Select the appropriate fields and enter credentials page is displayed.

8 Click and drag the onto the appropriate boxes, and then click and drag the onto *OK*. This ensures that all fields are active and can be identified by the wizard.

9 Click *Next.* The Name the application definition page is displayed.

10 Select the name of the application definition created initially for the application's logon (recommended), then click *Finish*. The Add Application Wizard updates the application definition and closes.

12.2 Responding to Application Messages

When you build a SecureLogin application definition for an application, it is important to respond to any messages that the application generates. Actions for each of these messages should be included in the application definition to ensure that SecureLogin responds appropriately.

- Section 12.2.1, "Changing an Application Definition to Respond to a Change Successful Message," on page 121
- Section 12.2.2, "Changing an Application Definition to Respond to a Login Successful Message," on page 121
- Section 12.2.3, "Changing an Application Definition to Respond to a Login Failure Message," on page 121

12.2.1 Changing an Application Definition to Respond to a Change Successful Message

After a password has been changed successfully, a Change Successful message appears in many application logins. Using the Add Application Wizard, you can change your application definition to respond to this event by clearing the application message and updating your SecureLogin stored credentials.

NOTE: Ensure that the Change Successful message is displayed so that the Add Application Wizard can identify it.

12.2.2 Changing an Application Definition to Respond to a Login Successful Message

---

NOTE: Ensure that the Login Successful message is displayed so that the Add Application Wizard can identify it.

---

1. In the notification area, right-click the Novell SecureLogin icon, then select *Add Application*. The Welcome to SecureLogin page is displayed.
2. Click *Next*. The Single sign-on enable an application page is displayed.
3. Select *Windows Application*, then click *Next*. The Single sign-on enable a Windows application login or message box page is displayed.
4. Click and drag the icon to the application's Change Password dialog box title bar. The Select window function page is displayed.
5. Click *Next.* The Select the appropriate fields and enter credentials page is displayed.
6. Click and drag the *Message Text* icon to the message text on the message.
7. Click and drag the *OK* Button icon to the *OK* on the message.
8. Click *Next*. The Name the application definition page is displayed.
9. Select the name of the application definition created initially for the application's login (recommended). Click *Finish*.

   The Add Application Wizard updates the application definition and closes.

12.2.3 Changing an Application Definition to Respond to a Login Failure Message

If an error occurs during login (for example, a credential is incorrect), the Login Failure message appears. Using the *Add Application* Wizard, you can change the application definition to respond to these events and update your SecureLogin stored credentials.

NOTE: Ensure that the Login Failure message is displayed so that the *Add Application* Wizard can identify it.

---

1. On the notification area, right-click the Novell SecureLogin icon, then select *Add Application*. The Welcome to SecureLogin page is displayed.
2. Click *Next*. The Single sign-on enable an application page is displayed.

3. Select *Windows Application*, then click *Next*. The Single sign-on enable a Windows application login or message box page is displayed.
4. Click and drag the  icon to the application's Change Password dialog box title bar. The Select window function page is displayed.
5. In the drop-down list, select *Incorrect Password Message*.
6. Click *Next*. The Select the appropriate fields and enter credentials page is displayed.
7. Click and drag the Message Text  icon to the message text on the message.
8. Click and drag the *OK* Button  icon to the *OK* on the message.
9. Click *Next*. The Name the application definition page is displayed.
10. Select the name of the application definition created initially for the application's login (recommended).

    Click *Finish*.

    The Add Application Wizard updates the application definition and closes. The next time the user logs on incorrectly, an error message appears.

NOTE: If the application returns different messages for similar errors (for example, different messages for an incorrect username or password), you should configure the *Add Application* Wizard for one message. Additional messages require editing the application definition using the `DisplayVariables` command.

Distributing Configurations

13

This section provides information on the following:

13.1 About Distributing Configurations

SecureLogin preferences, application definitions, password rules, and credentials are collectively the SecureLogin configured user environment. You can deploy and maintain this environment at all object levels, including by file import or backup to stand-alone users and through Group Policy Objects in Active Directory networks.

A single sign-on environment that is configured at the container, organizational unit, or Group Policy level is inherited by all associated directory objects in the hierarchy.

We recommend that you first enable applications for single sign-on locally, in a test user account, then copy to the container, OU or Group Policy level for mass deployment. This applies to all SecureLogin configurations, including password policies and preferences. Lower-level settings that you manually configure always override higher-level settings. Therefore, configuration at the user object level overrides all higher level configuration settings. You can manually disable inheritance by selecting *Yes* next to *Stop walking here* in the Preferences Properties table.

13.2 Distributing Configurations Within Directory Domains

There are two options for distributing the single sign-on-configured environment within the domain:

- **Corporate Redirection:** Specifies the object from which the selected object will inherit its SecureLogin configuration settings.
- **Copy SecureLogin Configuration:** Replicates and stores the SecureLogin environment from one directory object to another.

Choose the appropriate option based on the additional information in the following table:

Table 13-1 SecureLogin Configuration Options

If	Then
◆ Multiple containers or organizational units require the same SecureLogin environment, and you want to manage configuration from one directory object. ◆ Inheritance from a higher level than the object selected for Corporate Redirection is not required. ◆ The container or OUs are on the same directory tree. We do not recommend using Corporate redirection across a LAN or WAN.	Click *Corporate redirection*.
◆ You want to distribute configurations within the same domain across a LAN or WAN. ◆ You want to quickly replicate a complete SecureLogin configuration environment from one object to another in the directory. ◆ You do not want to use XML files to distribute SecureLogin configuration data.	Click *Copy SecureLogin configuration*.

13.3 Setting Corporate Redirection

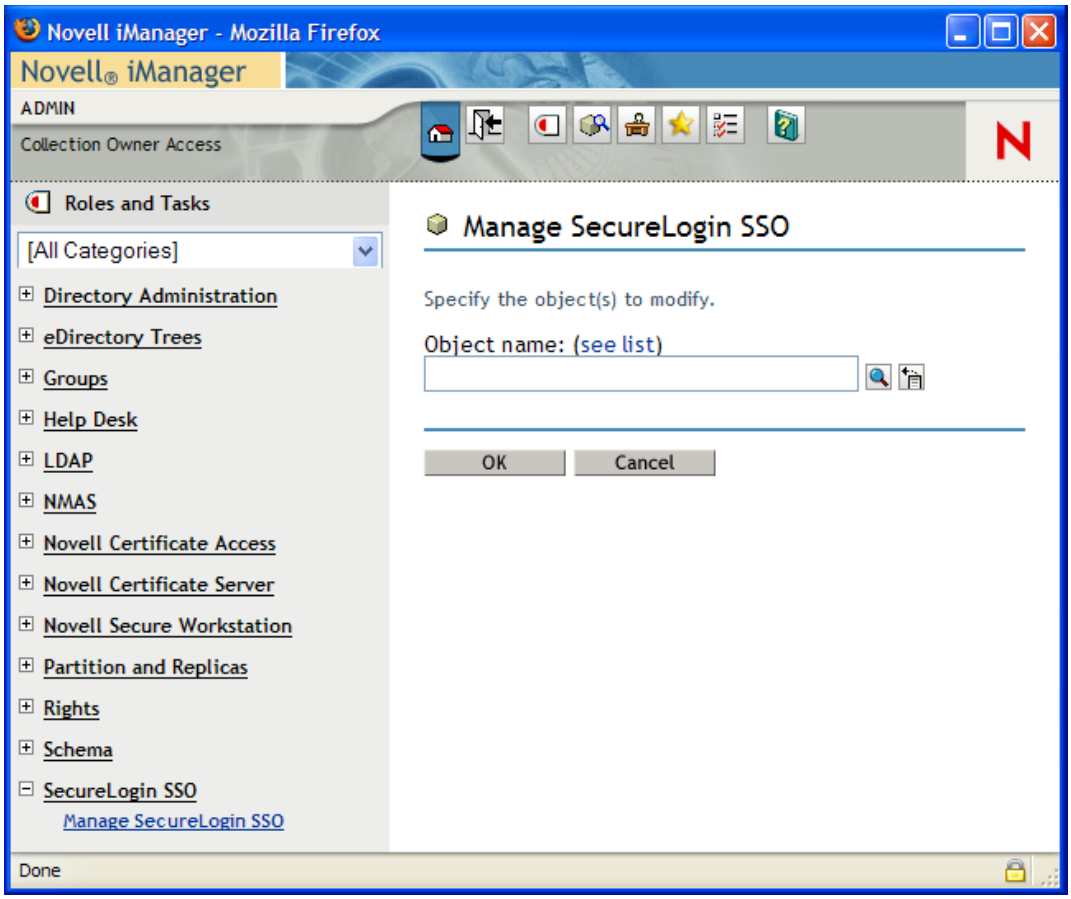
Before you set corporate redirection, Active Directory users and the Administrative Management utility must be active.

Corporate redirection functionality bypasses native directory inheritance by specifying the object from which the object inherits its SecureLogin configuration. Although inheritance is "redirected" to a specific object, such as a container or organizational unit, local user object settings continue to override the inherited settings.

NOTE: Inheritance of SecureLogin data using the *Corporate redirection* functionality stops at the container or organizational unit. Any settings, enabled applications, or password rules that are inherited by the container or organizational unit providing the SecureLogin environment are not inherited from a higher-level directory object.

1 Access the Administrative Management utility.

For information on accessing the Administrative Management utility see, Section 1.2, "Starting the Administrative Management Utilities," on page 14 and, or, Section 1.3, "Accessing the Single Sign-On Plug-In Through iManager," on page 15.



2 Click *Advanced Settings*. The Advanced Settings pane is displayed.

3 Specify the full distinguished name of the object in the *Corporate redirection* field.

NOTE: The full distinguished name is required to uniquely identify the container or organizational unit.

---

4 Click *Apply*.

5 Click *OK*.

IMPORTANT: Ensure that you do not overwrite administrator settings when distributing SecureLogin configuration environments. For example, if you set the preference *Allow users to view and change settings* to *No* and then copy this to the container or organizational unit as part of a SecureLogin environment, including the Administrator user object, the administrator cannot view or change SecureLogin settings because they reside in that organizational unit. To prevent this from happening, all administrator user objects should be located in a separate organizational unit, and administrator preferences should be manually configured.

13.4 Setting Corporate Redirection with eDirectory

Corporate redirection functionality bypasses eDirectory inheritance by specifying the object from which the object inherits its SecureLogin configuration. Although inheritance is redirected to a specific object, such as a container or organizational unit, local user object settings continue to override the inherited settings.

With the introduction of the eDirectory group membership feature in the Novell SecureLogin 6.1 release, you must make additional attribute assignments to the group objects. This is primarily required when users are using different administrative management utilities such as NWAdmin, ConsoleOne, or iManager.

IMPORTANT: This is required if you wish to use group management after upgrading to Novell SecureLogin.

To use the eDirectory group membership feature, you must run the ndsschema tool to correctly set the group, user, and container assignments before upgrading to Novell SecureLogin 6.1 SP1.

You can resolve this in one of the following ways:

- Run the ndsschema tool to assign the necessary rights and attributes or schema assignments to the group objects.
- Manage through iManager by running the Novell SecureLogin 6.1 SP1 plug-in.
- Section 13.4.1, “Configuring Groups Within eDirectory,” on page 126

13.4.1 Configuring Groups Within eDirectory

1. Access the Administrative Management utility.

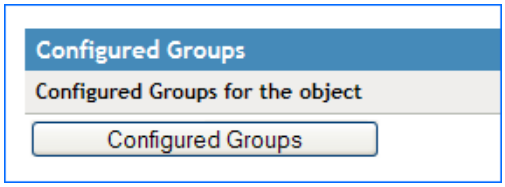
   For information on accessing the Administrative Management utility see, Section 1.2, “Starting the Administrative Management Utilities,” on page 14 and, or, Section 1.3, “Accessing the Single Sign-On Plug-In Through iManager,” on page 15.

2. Specify the distinguished name of the container object you want to modify.

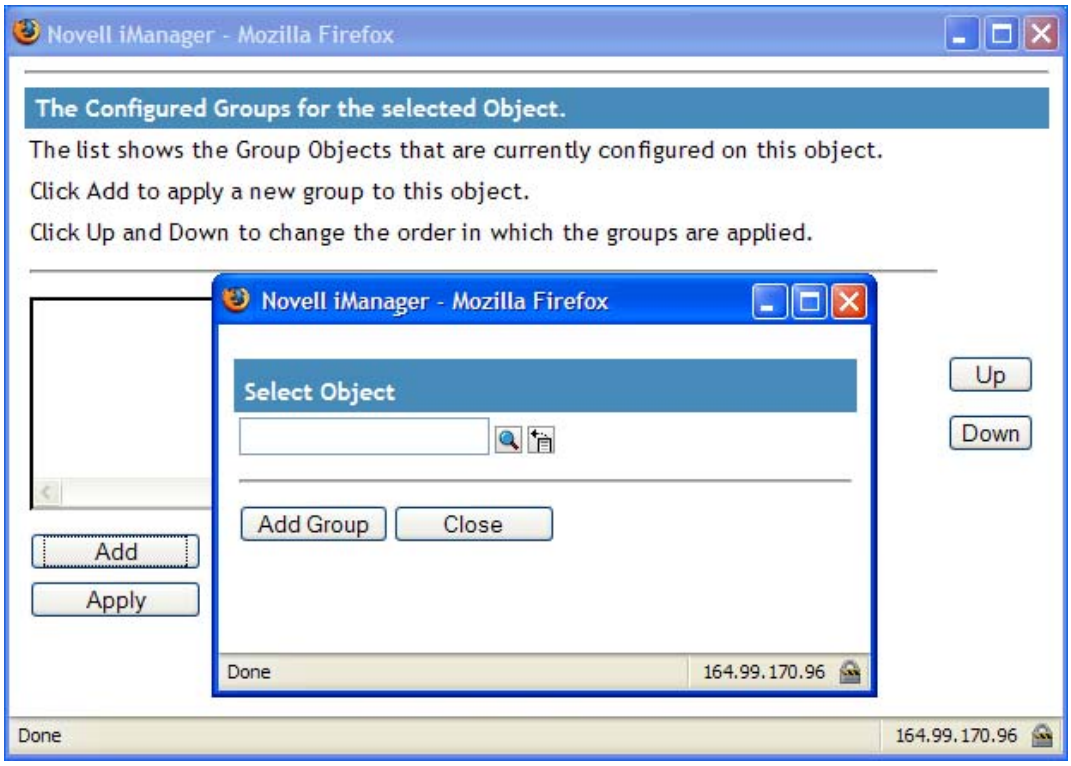
   NOTE: You can modify only container object to configure group.

   ---

3. Select *Advanced Settings* > *Configured Groups*. The Group Configuration dialog box is displayed.

4. Click *Add*. The Adding a group dialog is displayed.
5. Provide the distinguished name of the group object.



6 Click *OK* to add the new group object. The Group Configuration dialog is displayed.

Use the *Up* and *Down* options to promote or demote the order in which the group policies are applied.

Within the Group Configuration, the higher group takes precedence.

Configured groups can only be set against containers like O and OU and not set against a user object. In such a case, contrary to the earlier statement, the higher container takes the lower precedence.

NOTE: After you have configured single sign-on settings for Dynamic Group, the configuration is not reflected iManager for member users.

However, the configured settings are available in the Client when Novell SecureLogin is launched.

13.5 Copying a Configuration Across Organizational Units

You can copy an object's SecureLogin configuration to another object from the Distribution pane in the Administrative Management utility. This functionality replicates the SecureLogin configuration internally in the same directory tree.

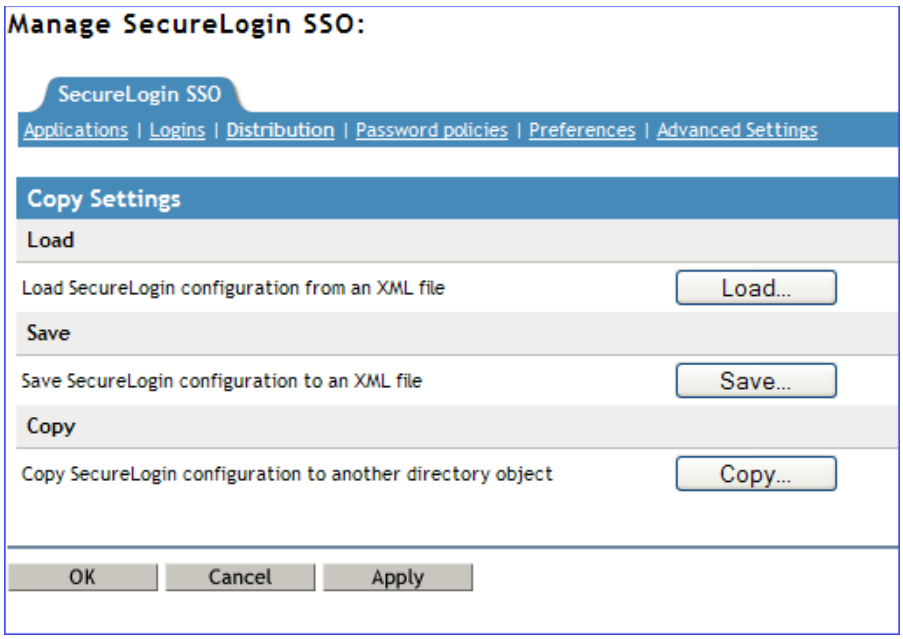
NOTE: In the following example, the Development organizational unit SecureLogin environment is copied to the Finance organizational unit.

---

1 Access the Administrative Management utility.

For information on accessing the Administrative Management utility, see Section 1.2, "Starting the Administrative Management Utilities," on page 14 and, or, Section 1.3, "Accessing the Single Sign-On Plug-In Through iManager," on page 15.

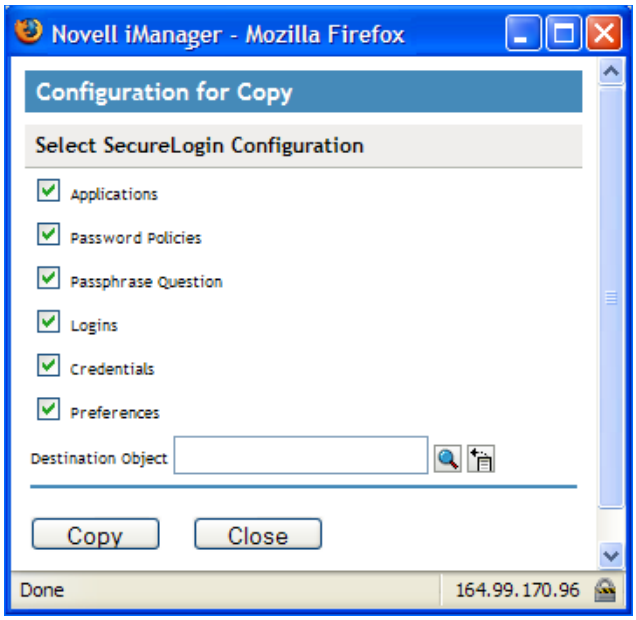
2 Click *Distribution*. The Distribution pane is displayed.



3 Click *Copy*. The Copy dialog box is displayed.

4 Under *Select SecureLogin Configuration*, select or clear the appropriate check boxes.

Configuration	Function
Applications	Copies, exports, or imports all configured application definitions, as displayed in the Applications pane.
Credentials	Copies, exports, or imports all credentials as displayed in the Logins pane, excluding passwords for copy settings and uninterrupted export/import.
Password Policies	Copies, exports, or imports password policies as displayed in the Password Policies Properties table
Preferences	Copies, exports, or imports all preferences manually set in the Preferences pane.
Active Passphrase Question	Provides users with a selection of passphrase questions. This option copies, exports, or imports only the passphrase question the user has responded to.



5 In the *Destination Object* drop-down list, click the name of the object or type the full distinguished name in the box.

6 Click *Copy*.

If a predefined application or an application definition currently exists in the destination object, a confirmation message appears. It confirms or rejects the overwriting of the imported data.

7 Click *Yes* or *No* as required.

The selected SecureLogin configuration is copied across to the destination user object, organizational unit or container. A confirmation message appears, advising what information has been loaded to the destination object.

8 Click *OK*.

13.6 Creating an Active Directory Group Policy

13.6.1 Group Policy Object Support

Using Group Policy object support, you can manage SecureLogin users in Active Directory users at the container, OU, and user object levels.

Group Policy object support is useful for organizations with flat directory structures where a more granular approach is required when applying settings, policies, and application definitions for users. For example, applying a group policy for a global marketing group in a worldwide organization.

Several group policies can be defined and applied to any user, group, or container at the directory level. These different policies are then applied to a specific user object or container or organizational unit through the inheritance process.

To limit network traffic during the Group Policy object synchronization, Novell SecureLogin leverages an existing Microsoft Windows feature to specify policy settings that are updated when the group policy object changes.

In the SecureLogin GPextensions in the Windows Registry, set the `NoGPOListChanges` key to 1.

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVerison\Winlogon\GPExtensions\<Class-ID>
```

For more information on Microsoft Windows Group Policy configuration, see the Microsoft Web site. (http://www.microsoft.com/windows/windows2000/en/advanced/help/ComputerADM.htm)

For information on the Registry `NoGPOListChanges` setting, see the Microsoft Web site. (http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/regentry/93807.mspx?mfr=true)

13.6.2 Group Policy Management Console Support

In Novell SecureLogin, you can see the resultant set of single sign-on policy settings that apply to a particular user object when multiple SecureLogin group policies and organizational unit or user object setting are applied through the Microsoft's Group Policy Management Console (GPMC), which now includes support for Resultant Set of Policy (RSOP).

NOTE: The GPMC must be installed on the administrative workstation where you want to see the resultant set of policies.

Resultant Set of Policy Settings

The Resultant Set of Policy (RSOP) is a feature of a group policy that makes the implementation, troubleshooting, and planning of group policies easier and allows you to plan how the group policy changes might affect a targeted user or computer or remotely verify the policies under effect on a specific computer.

When multiple group policy objects are applied to a given user or computer, the policy can often contain conflicting policy settings. For most policy settings, the final value of the setting is set only by the highest precedent Group Policy object that contains that setting.

RSOP assists directory administrators to understand and identify the final set of policies that are applied as well as settings that did not apply as a result of policy inheritance.

In this version of Novell SecureLogin, you can see the final SecureLogin settings that apply to a user when he or she starts Novell SecureLogin. You have the ability to do the following:

- Retrieve the policy applied to the user object in the Microsoft Management Console.
- Retrieve the policy applied to the user object in the SLManager.
- Define from which policy the setting is inherited.

13.6.3 Adding or Editing a Group Policy Object

Policy settings are stored in Group Policy object settings for each Group Policy object can be edited using the Group Policy object editor from Microsoft's GPMC.

The group policy functionality is enabled during the installation of SecureLogin in either Microsoft Active Directory mode. For more information see, "Installing in a Microsoft Active Directory Environment" in the *Novell SecureLogin 6.1 SP1 Installation Guide*.

When you define a SecureLogin Group Policy Object, users can use the GPMC to add this group policy or edit and configure the SecureLogin settings.

13.6.4 Installing the GPMC Plug-In

With the Microsoft's GPMC plug-in, you can manage core aspects of Group Policy object across enterprises.

For Microsoft Vista customers, the GPMC snap-in is already integrated in to the operating system.

Existing Windows XP and Server customers can download the `gpmc.msi` installer package at the Microsoft Web site (http://www.microsoft.com/windowsserver2003/gpmc/default.mspx).

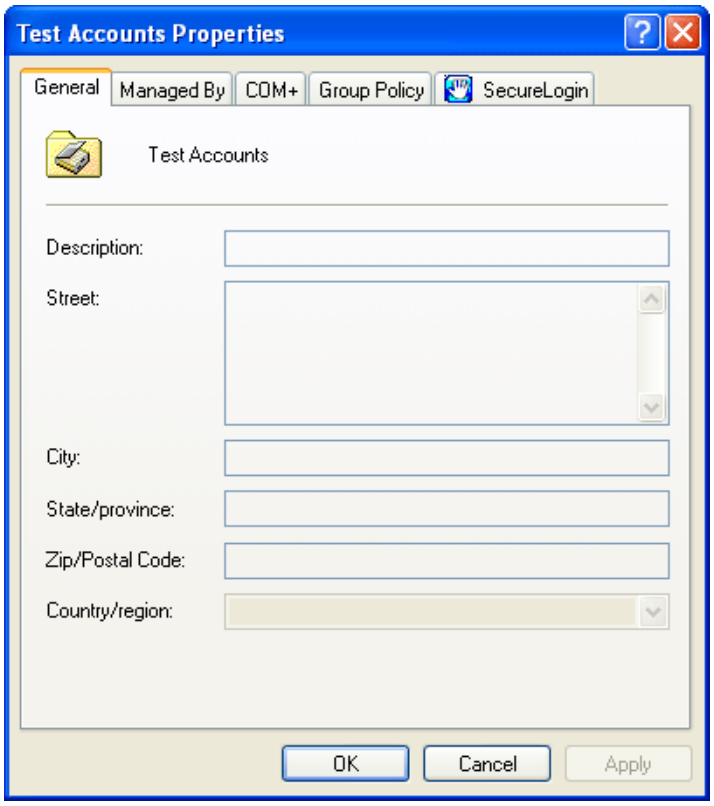
Installing the Microsoft GPMC plug-in simply involves running the gpmc.msi installer package.

NOTE: After installation, the *Group Policy* tab that previously appeared on the Property pages of sites, domains, and organizational units in the Active Directory plug-in is updated to provide a direct link to GPMC. The functionality that previously existed on the original *Group Policy* tab is no longer available because all functionality for managing a Group Policy is available through the GPMC plug-in.

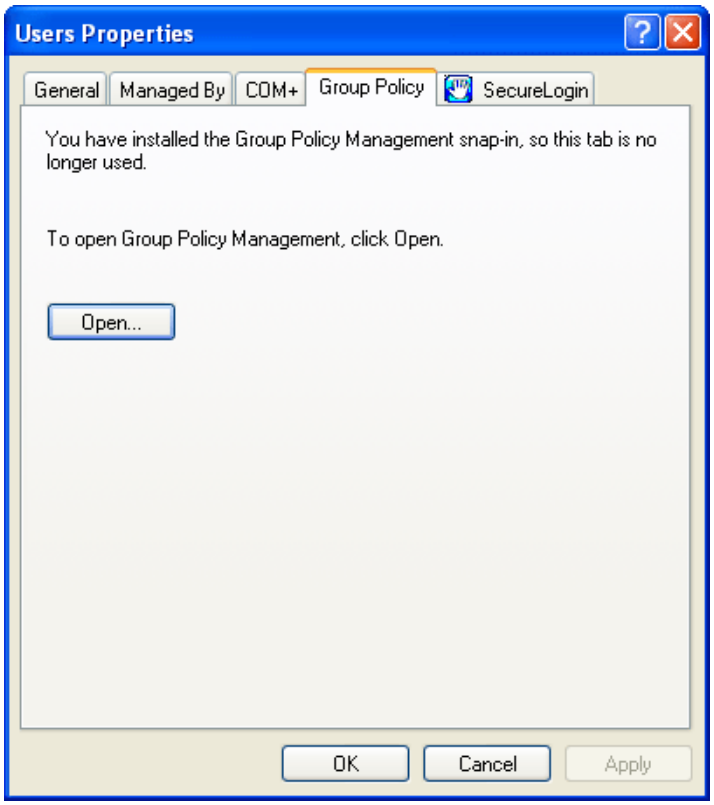
Managing Group Policy Objects through the GPMC

Use any of the following methods to open the GPMC plug-in directly:

- Click *Start* > *Programs* > *Administrative Tools* > *Active Directory Users and Computers*. The Active Directory Users and Computers page is displayed.
- In the navigation tree, right-click the appropriate organizational unit, then click *Properties*. The selected organizational unit page is displayed.



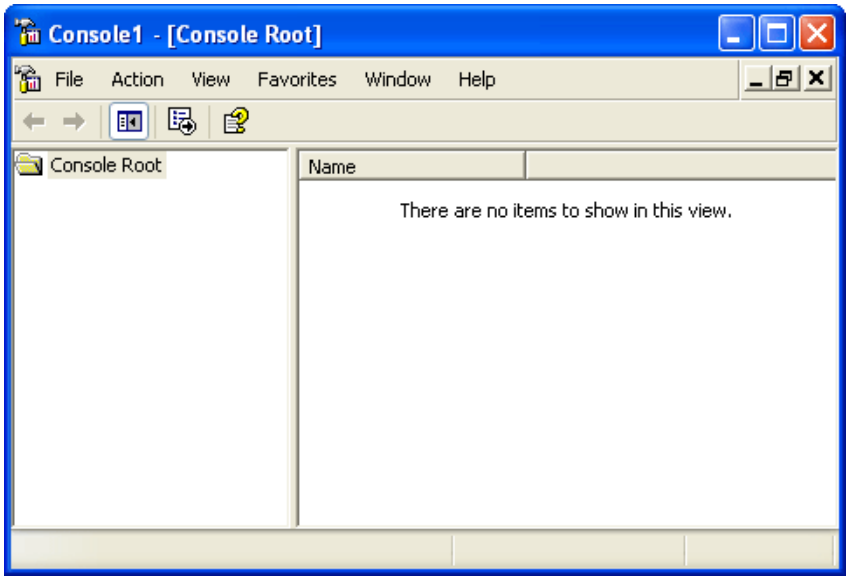
- Click *Group Policy*, then click *Open*.



- Click *Start* > *Programs* > *Administrative Tools* > *Group Policy Management*.
- Click *Start* > *Run*. The Run page is displayed.

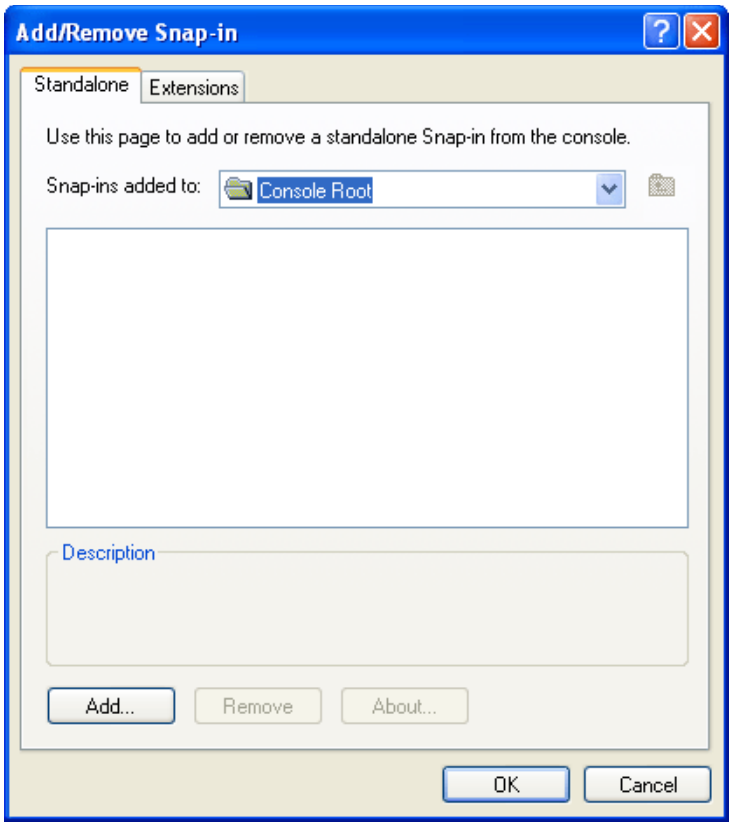
1 At *Open*, type `mmc`.

2 Click *OK*. The Management Console is displayed.

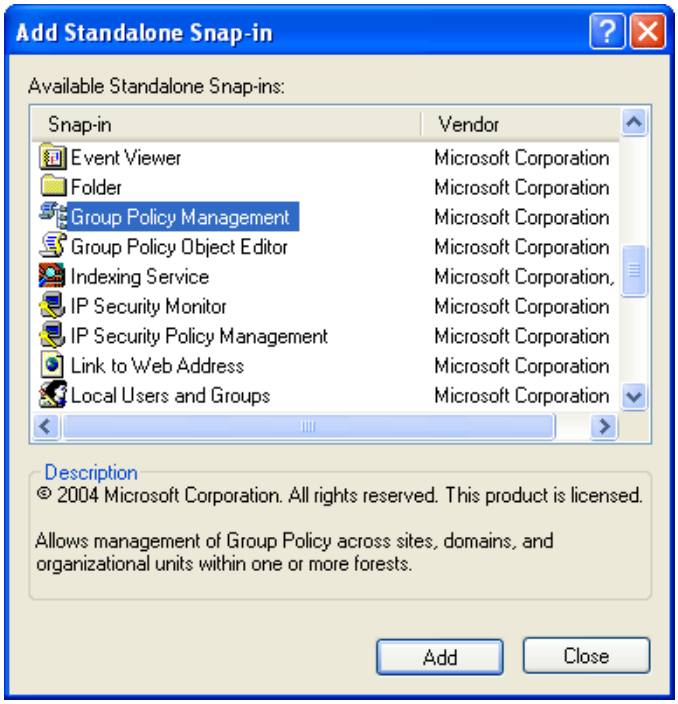


3 Click *File*.

4 Click *Add/Remove Snap-in*. The *Add/Remove* page is displayed.

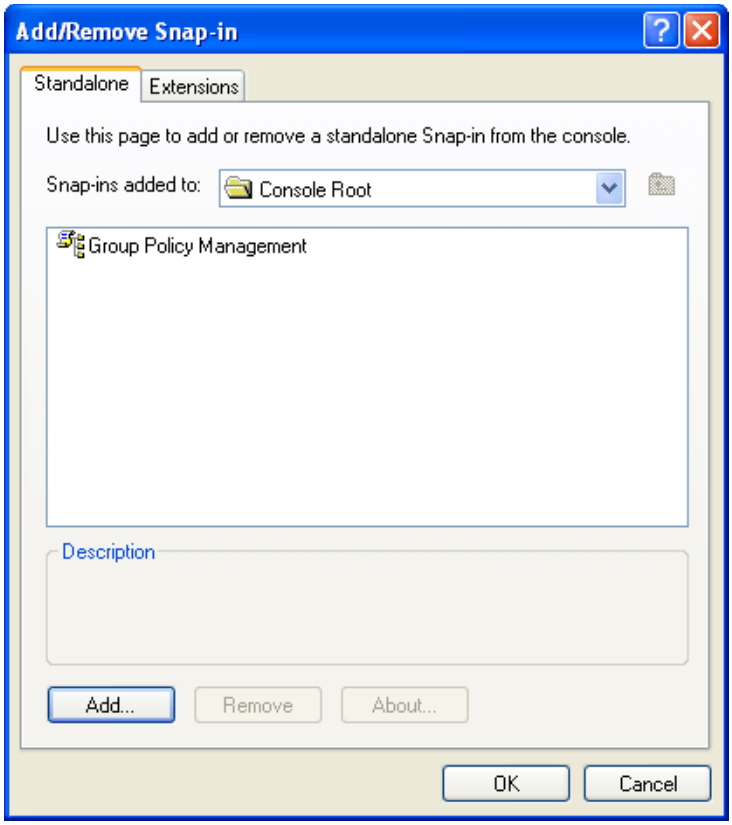


5 Click *Add*. The Add Standalone Snap-in page is displayed.



6 Select *Group Policy Management* and then, click *Add*.

7 Click *Close*. The Add Standalone Snap-in page is displayed.



8 Click *OK*. The Group Policy Management page is displayed.

NOTE: When you launch the GPMC for the first time, it loads the forest and domain containing the user object logged in to the computer. You can then specify the forest and domain to be displayed.

When you close the GPMC, it automatically saves the last view and returns that view the next a user opens the console.

13.6.5 Retrieving a Policy Applied to the User Object in GPMC

The definition of the Group Policy Objects are defined by the administrator at the directory level, so changes can now be seen immediately at the OU, container or user object level, depending on the level where the group policies have been applied and the SecureLogin preferences applied.

These settings must follow the rules already defined of inheritance and precedence:

- The *Stop walking here* preference
- The *Corporate Redirection* setting
- The Group Policy object settings and their priorities
- The directory hierarchy settings

The precedence rules are respected and follow the rules already defined:

- The deepest object in the tree has the precedence over any other higher-level object
- The group policies have the lower precedence than all OUs and User objects.

As a consequence of all these processes, the administrator can now see the resultant set of the policies in the user object either through MMC interface or administrative management utilities.

The resultant set of policies are displayed in the bottom left hand corner of the SecureLogin Administration Management utility. They show from which Group Policy the current setting has been inherited.

NOTE: The retrieval of all SecureLogin configuration information is subject to both SecureLogin and native Directory access controls. In the unlikely circumstance that the user has rights to read a Group Policy object but the administrator does not, this system displays incorrect effective configuration information. This is because the administrator simply cannot access the same information as the user, and any mechanism for allowing this would introduce a security problem.

In this specific configuration, if SecureLogin has no way to retrieve the exact policy applied to the user object, then a message is displayed indicating that the information displayed does not correspond to the resultant set of policies applied to this user object. The message *RSOP not available* is displayed in the bottom left side of the Administration Management console.

13.6.6 Retrieving a Policy Applied to the User Object in SLManager

Because the definition of the Group Policy objects are performed by you at the directory level, any changes are now seen immediately at the OU, container, or the user object level, depending on the level where the group policy is applied and the Novell SecureLogin preferences is applied.

Exporting and Importing Configurations

14

The export and import functionality of SecureLogin creates an XML file, internal or external to the directory. You can distribute and back up this file across directory types, servers, domains, containers, group policies, organizational objects, and user objects.

SecureLogin's export and import functionality creates and XML file, internal or external to the directory. You can distribute and back up this file across directory types, servers, domains, containers, group policies, organizational objects, and user objects.

You can also encrypt and password-protect or digitally sign the exported files to ensure the information is secure. Alternatively, an unencrypted file can also be created for unrestricted distribution.

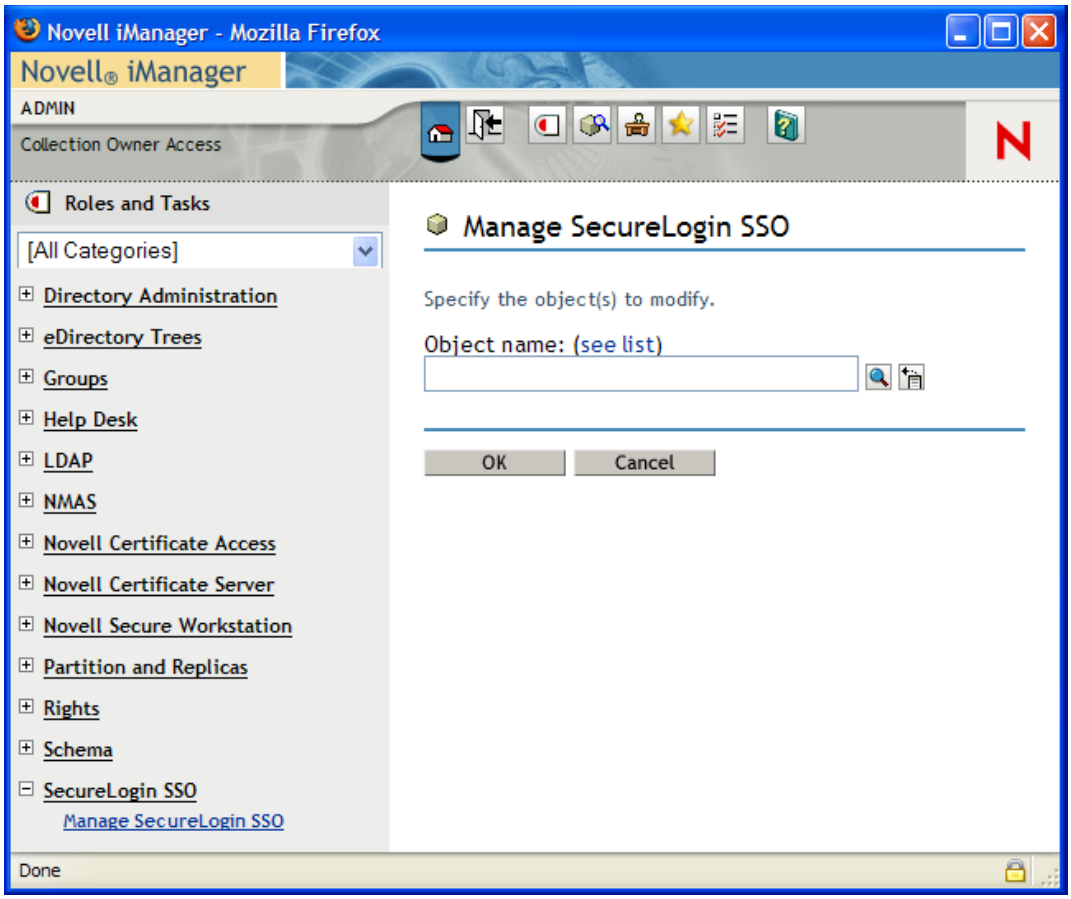
You can export or import the following XML file types:

- Unencrypted.
- Encrypted and password-protected.

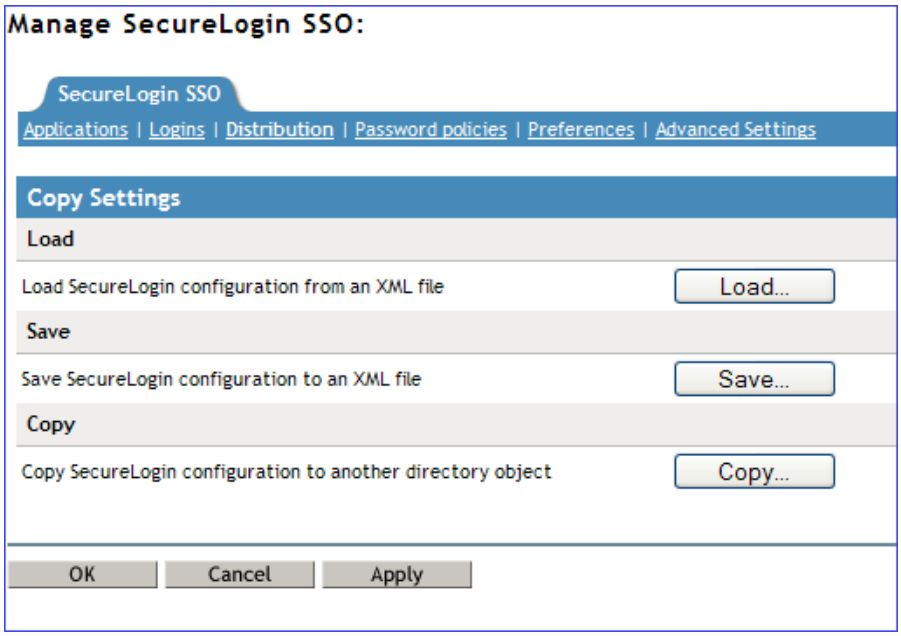
14.1 Exporting XML Settings

To export XML settings:

1. Log in to iManager.
2. Select *Securelogin SSO* > *Manage Securelogin SSO*. The Manage SecureLogin SSO page is displayed.
3. In the object field, specify your object name, then click *OK*.

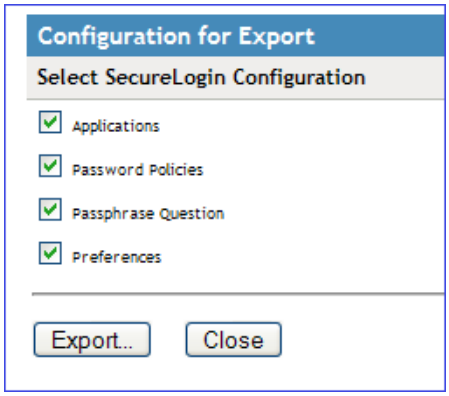


4 Click *Distribution*. The distribution details are displayed.



5 Click *Save*. The Configuration for Export dialog box is displayed.

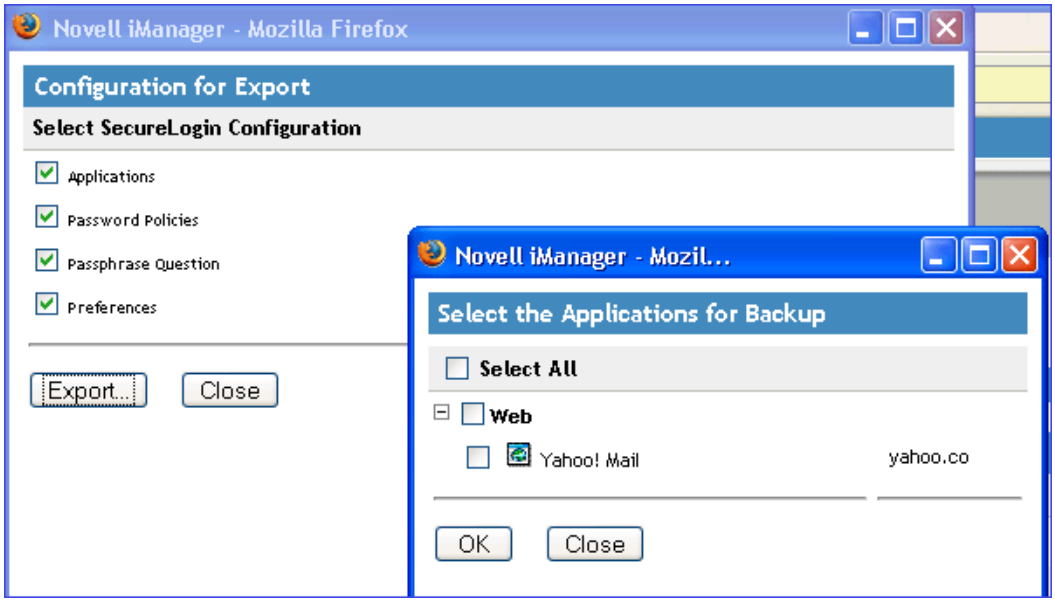
6 Under *Select SecureLogin Configuration*, select the appropriate text boxes.



Configuration	Function
Application	Copies, exports, or imports all configured application definitions as displayed in the *Application* pane.
Credentials	Copies, exports, or imports all credentials as displayed in the *Logins* pane, excluding passwords for copy settings and unencrypted export or import.
Password Policies	Copies, exports, or imports password policies as displayed in the *Password Policies Properties* table.
Preferences	Copies, exports, or imports preferences manually set in the *Preferences Properties* tables.

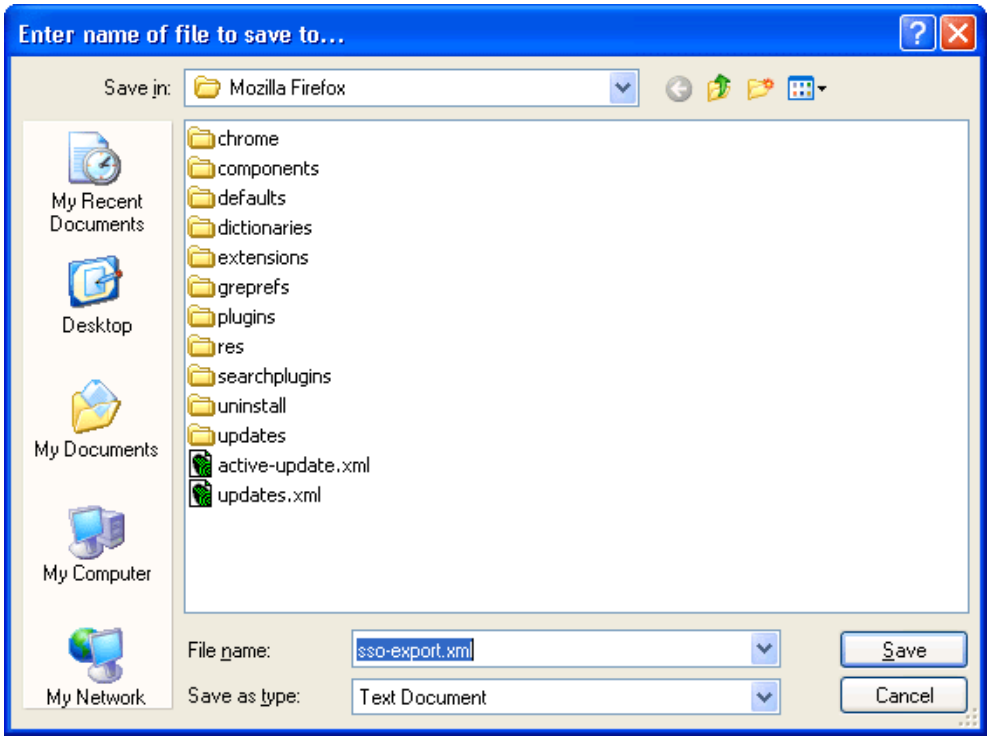
7 Click *Export*. The Select the Applications for Backup page is displayed.

8 Select the applications you want to backup.



9 Click *OK*. The Save File As dialog box is displayed.

10 Provide a name to the file, select the file location, and click *Save*.

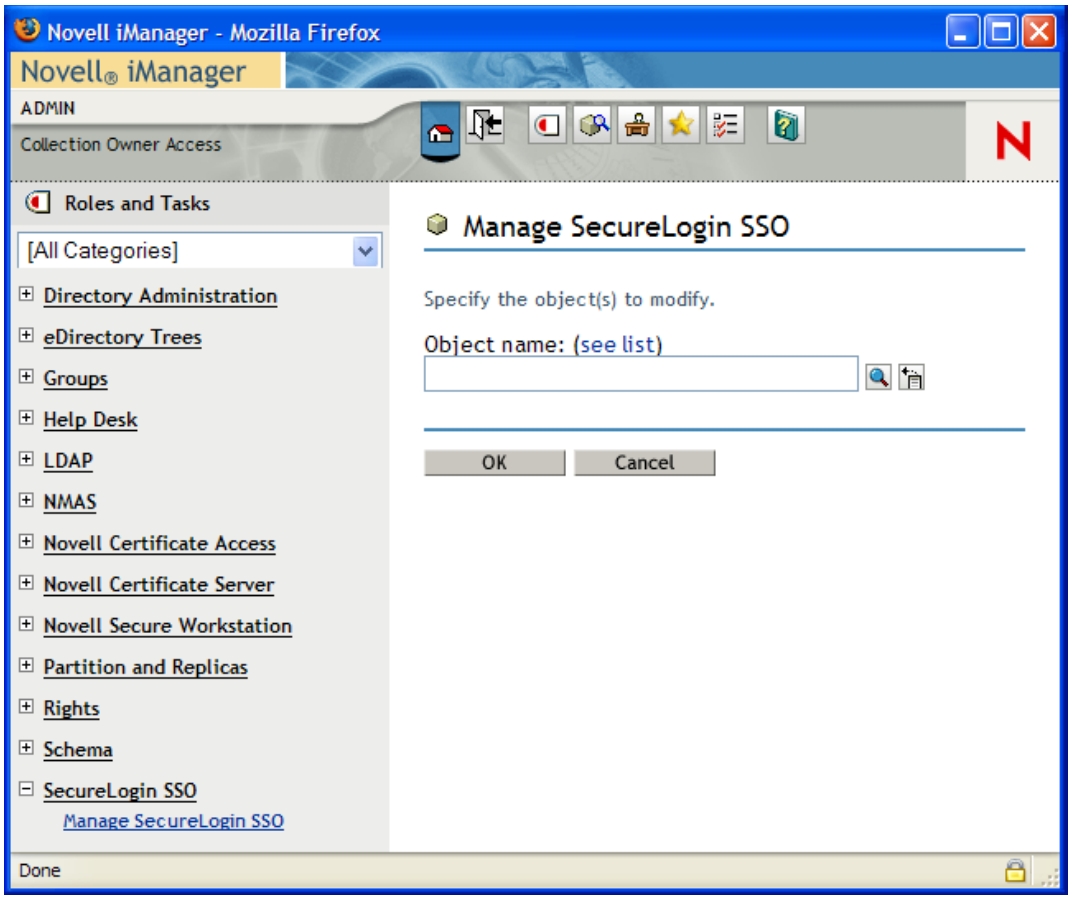


14.2 Importing XML Settings

To import XML settings:

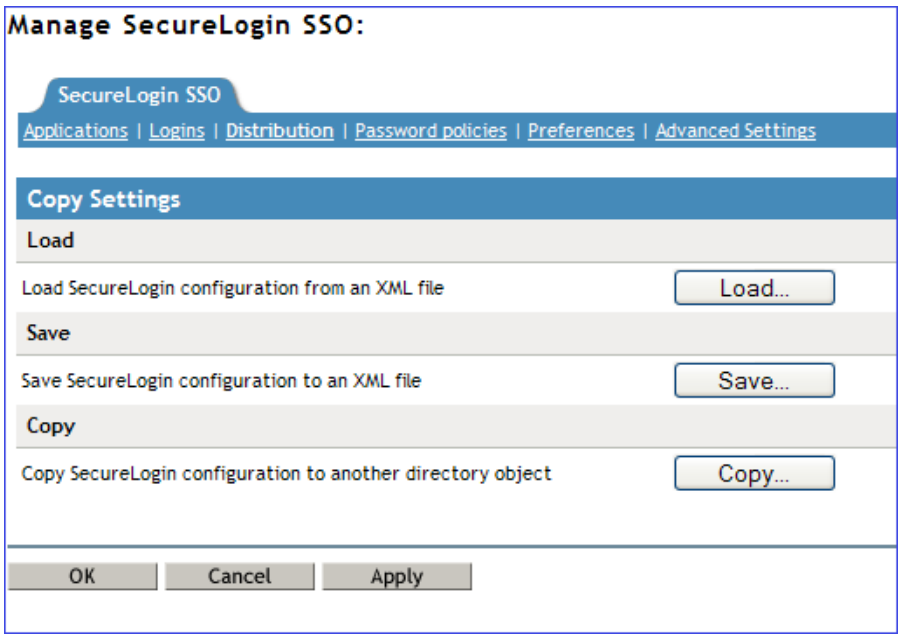
1 Log in to iManager.

2 Select *Securelogin SSO* > *Manage Securelogin SSO*. The Manage SecureLogin SSO page is displayed.

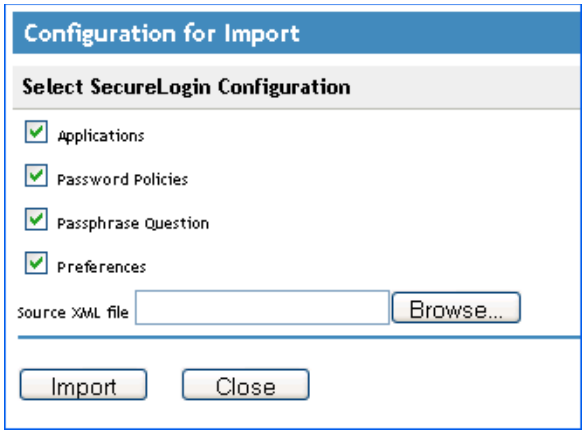


3 In the object field, specify your object name, then click *OK*.

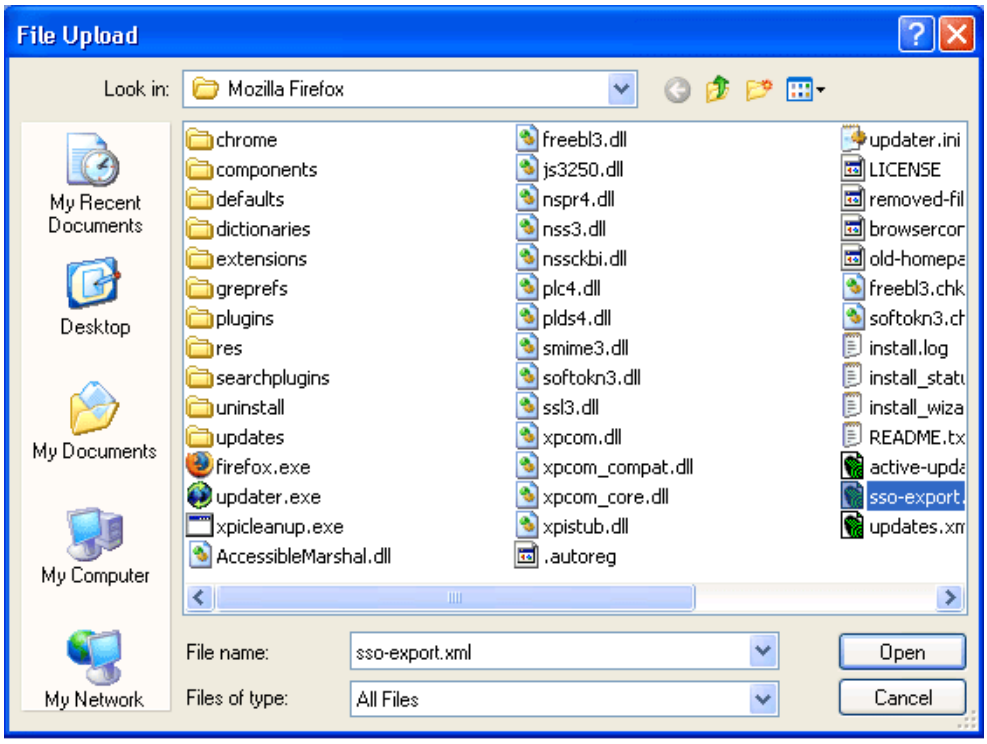
4 Click *Distribution*. The Distribution details are displayed.



5 Click *Load*. The Select SecureLogin Configuration dialog box is displayed.



6 Browse to and select the exported XML file.



7 Click *Open* to select the file.

The selected predefined applications and application definitions are copied across to the receiving organizational unit or container.

The selected Securelogin configuration is copied across to the receiving object.

If predefined applications and application definitions currently exist in the receiving object, a confirmation message is displayed to confirm or reject overwrite with the imported data.

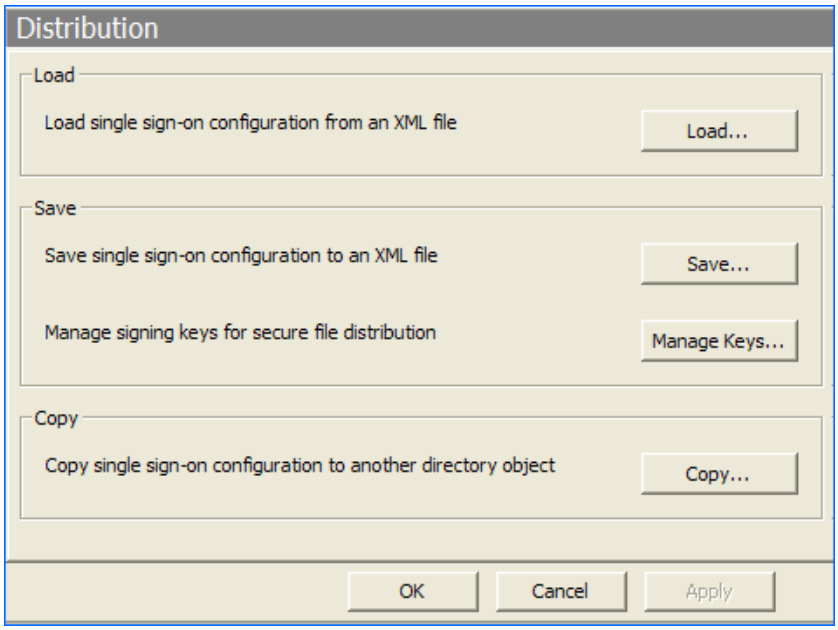
8 Click *Import* to confirm or click *Cancel* to reject overwriting with the imported data.

A SecureLogin message is displayed to confirm SecureLogin data is loaded.

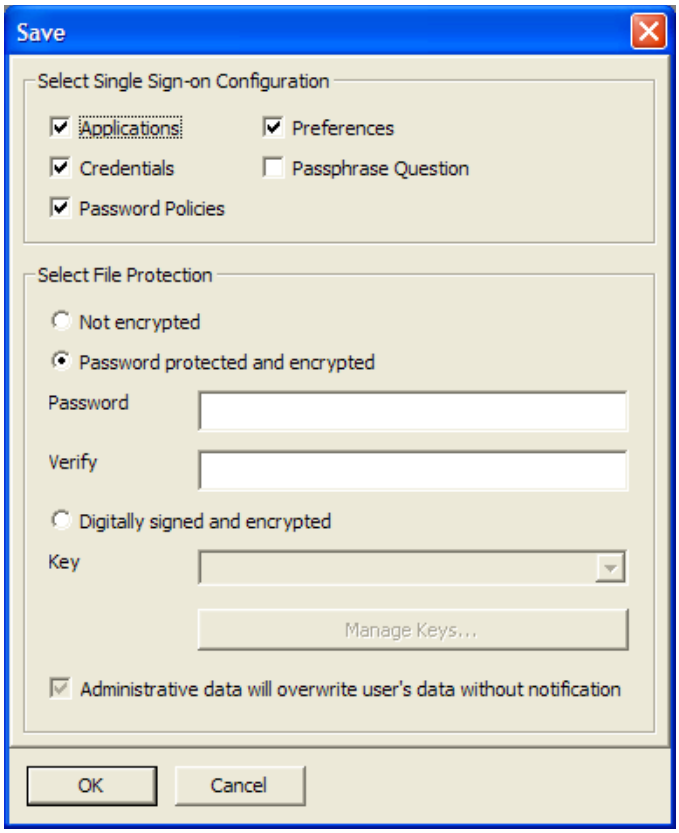
14.3 Exporting Single Sign-On Data in Encrypted XML Files

IMPORTANT: This option is available only through SecureLogin Manager.

1. Launch SecureLogin Manager.
2. In the object field, specify your object name, then click *OK*.
3. Click *Distribution*. The Distribution details are displayed.



4. Click Save. The save dialog box is displayed.



5 Select the appropriate options. The following table describes the options:

Configuration	Function
Application	Copies, exports, or imports all configured application definitions as displayed in the *Application* pane.
Credentials	Copies, exports, or imports all credentials as displayed in the *Logins* pane, excluding passwords for copy settings and unencrypted export or import.
Password Policies	Copies, exports, or imports password policies as displayed in the *Password Policies Properties* table.
Preferences	Copies, exports, or imports preferences manually set in the *Preferences Properties* tables.
Passphrase Question	Provides users with a selection of passphrase questions. This option copies, exports, and imports only those passphrase questions to which the user has responded.

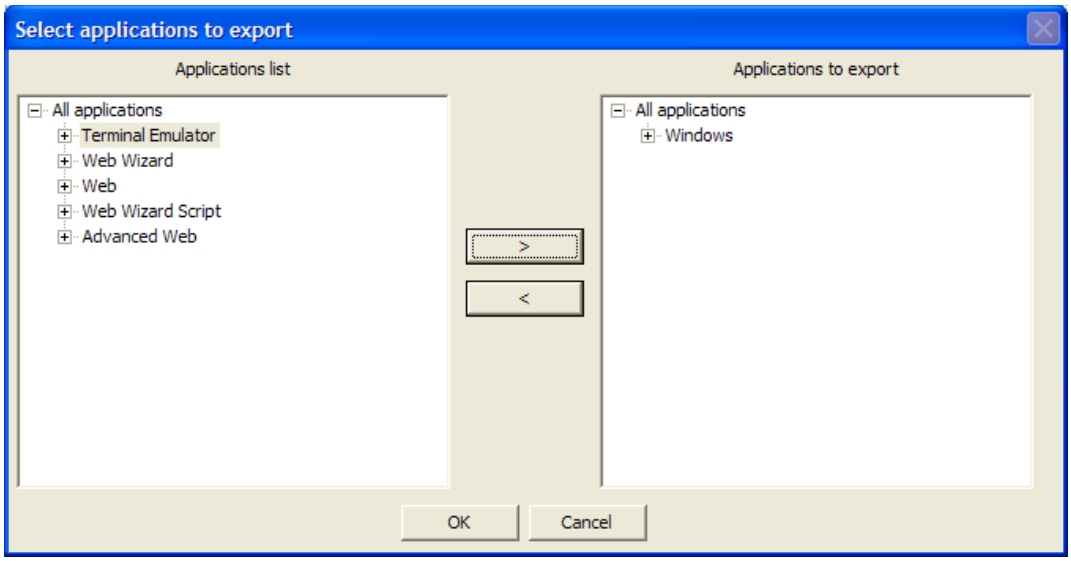
6 From *Select File Protection*, select *Password protected and encrypted*.

7 Specify the password in the *Password* field.

8 Re-specify the password in the *Verify* field.

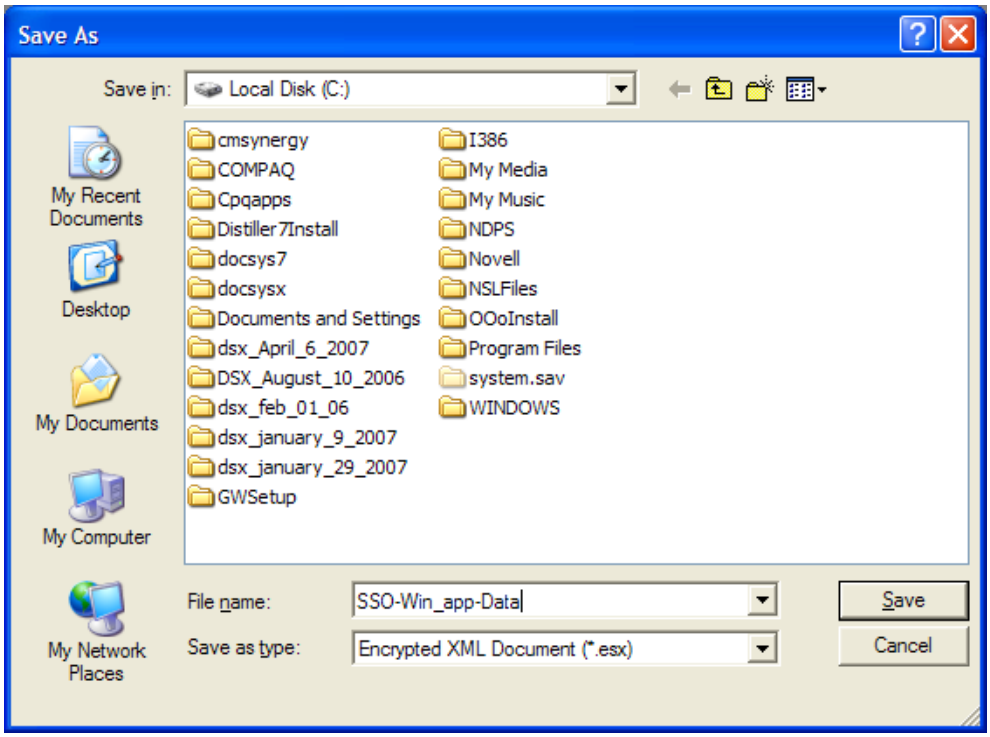
9 Click *OK*. The select application to export dialog box is displayed.

10 Select the applications to be exported, then click *OK*.



11 Select a location to save the file.

12 Specify a name for the file.



13 Click *Save*. The selected SecureLogin configuration is saved and a confirmation message appears indicating the information that is saved.

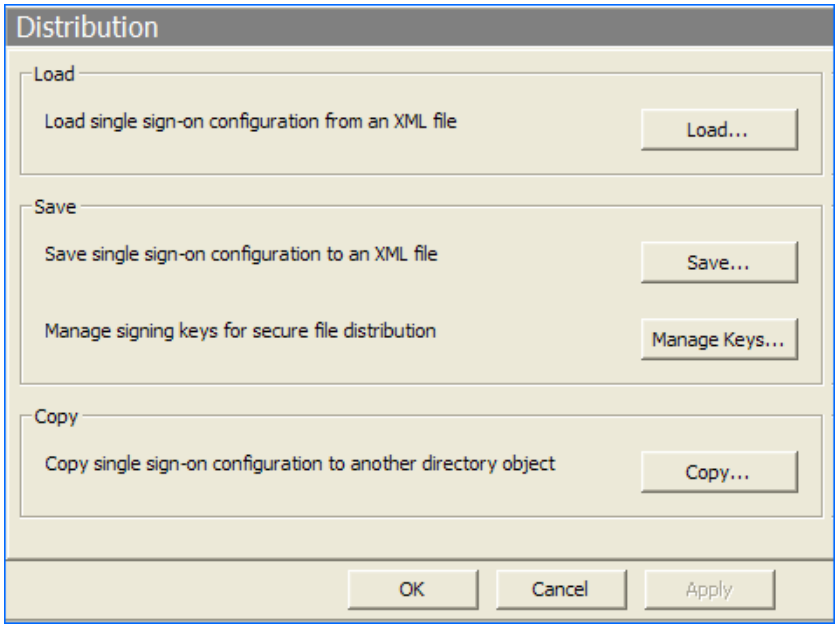
14 Click *OK*.

14.4 Importing Single Sign-On Data in Encrypted XML Files

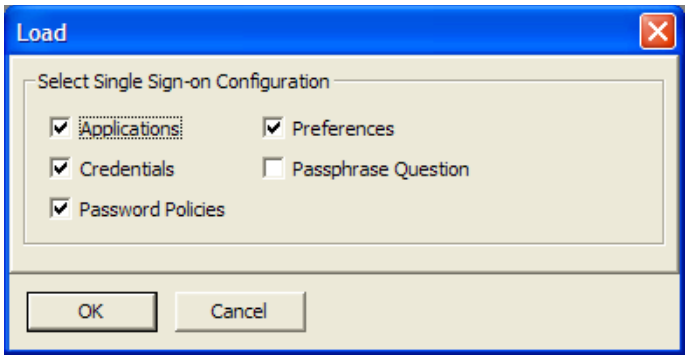
1 Launch SecureLogin Manager.

2 In the object field, specify your object name, then click *OK*.

3 Click *Distribution*. The Distribution details are displayed.



4 Click *Load*. The load dialog box appears.

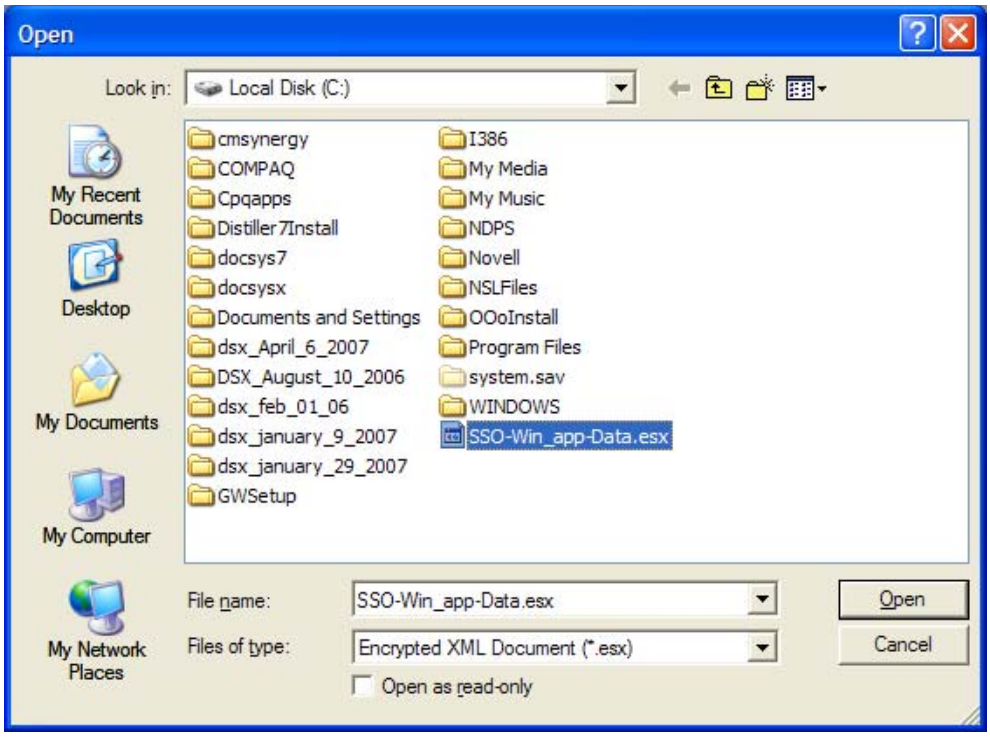


5 Select the required options. The following table helps you choose.

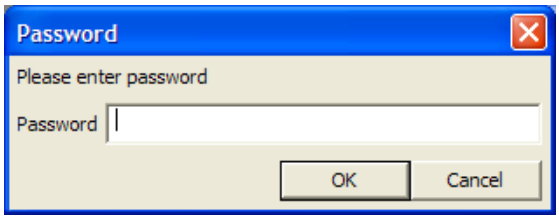
Configuration	Function
Application	Copies, exports, or imports all configured application definitions as displayed in the *Application* pane.
Credentials	Copies, exports, or imports all credentials as displayed in the *Logins* pane, excluding passwords for copy settings and unencrypted export or import.
Password Policies	Copies, exports, or imports password policies as displayed in the *Password Policies Properties* table.
Preferences	Copies, exports, or imports preferences manually set in the *Preferences Properties* tables.
Passphrase Question	Provides users with a selection of passphrase questions. This option copies, exports, and imports only those passphrase questions to which the user has responded.

6 Click *OK*. The open dialog box is displayed.

7 Select the exported encrypted file.

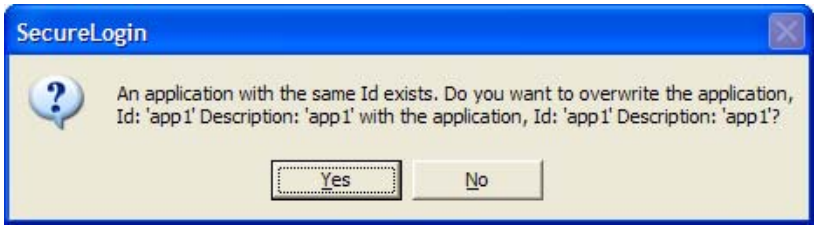


8 Click *Open*. The password dialog box is displayed.



9 Specify the password, then click *OK*.

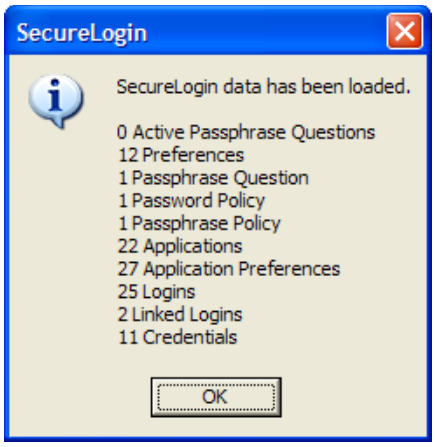
If a predefined application or an application definition currently exists in the destination object, you get a confirmation message appears for the applications.



Click *Yes* if you are sure that the imported application definition is preferred over the application definition currently stored in the user cache.

Click No to retain the application definition currently stored in the user cache.

10 If you click Yes, the configuration is copied across to the user object, organizational unit, or container. A confirmation message appears indicating that the information is copied to the destination object.



11 Click *OK*.

14.5 Creating a Signing Key for Secure Distribution

After you have configured and tested Novell SecureLogin in an user environment, you can create a digital signing key that is embedded in the distribution file (`.msi` file). You can distribute the file through a Web download or e-mail to the users. When users receive the file, they need to double-click the file to load to the local workstation. This updates the following:

- Preferences
- Application definitions
- Password rules
- Credentials

This is collectively known as the SecureLogin configured user environment and, is particularly designed for users who use Novell SecureLogin in standalone mode (such as mobile users) and those who do not frequently connect to the corporate network.

When a digital signing key is created, the key pair is randomly generated by the Novell SecureLogin to increase security.

To create a digital signing key:

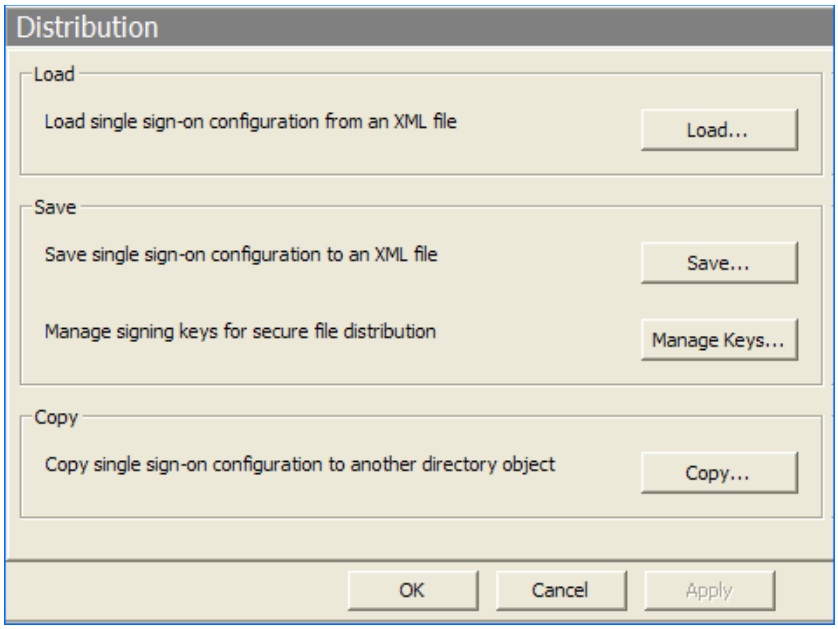
IMPORTANT: This feature is available only through SecureLogin Manager.

---

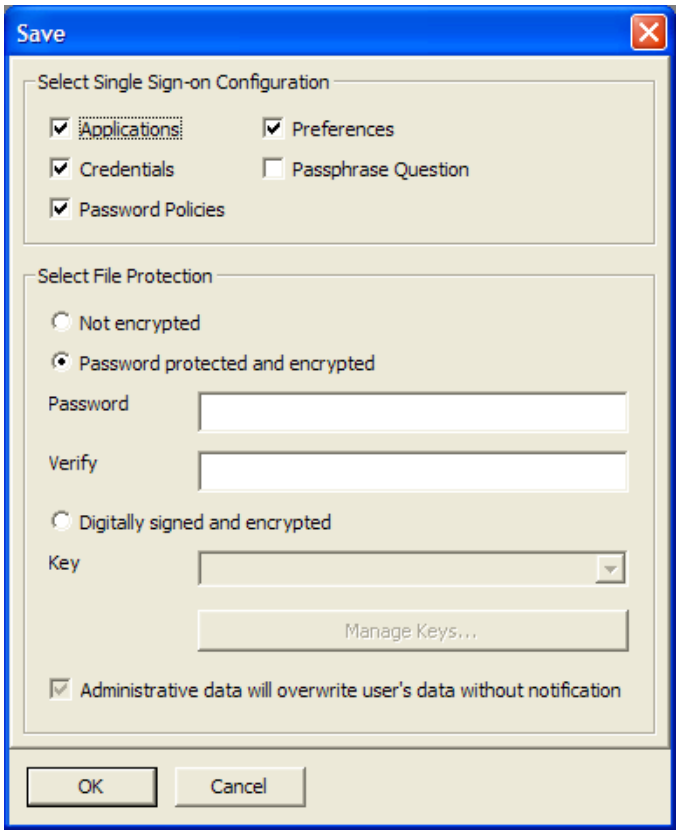
1 Launch SecureLogin Manager.

2 In the object field, specify your object name, then click *OK*.

3 Click *Distribution*. The Distribution details are displayed.



4 Click *Save*. The save dialog box is displayed.

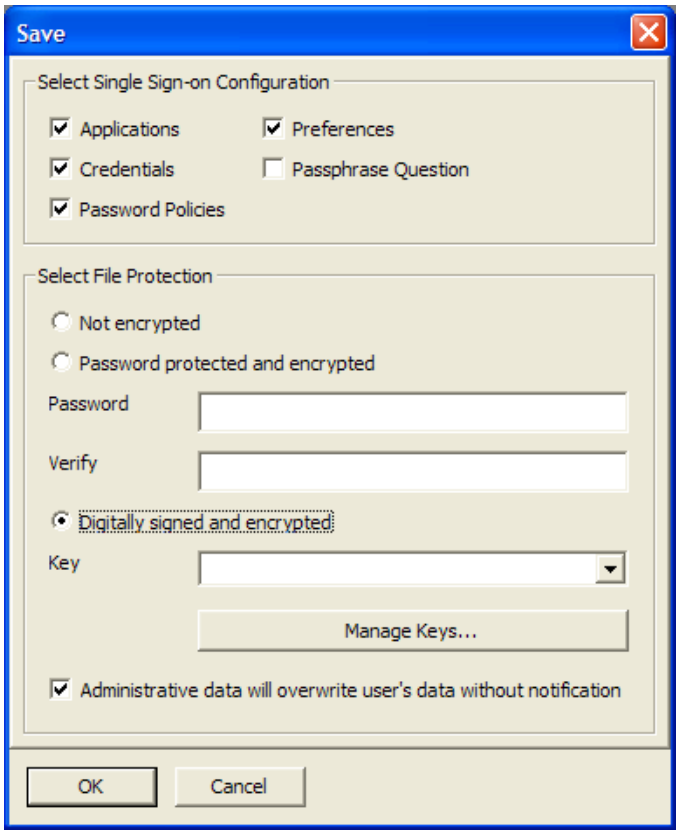


5 Select the required options.

6 Under *Select File Protection*, select *Digitally signed and encrypted*.

7 (Optional) Select Administrative data will overwrite user's data without notification.

If this option is selected, the users are prompted before overwriting any data with the configuration settings saved in the `.msi` file.



IMPORTANT: Selecting this option results in the user data being overwritten with the configuration setting in the `.msi` file for any items that are present in both the user's local configuration and the administrative configuration (`.msi` file).

For example, if a user have an application definition configured locally, and a predefined application definition is supplied in the `.msi` file, the `.msi` file application definition overwrites the user's application definition without notification.

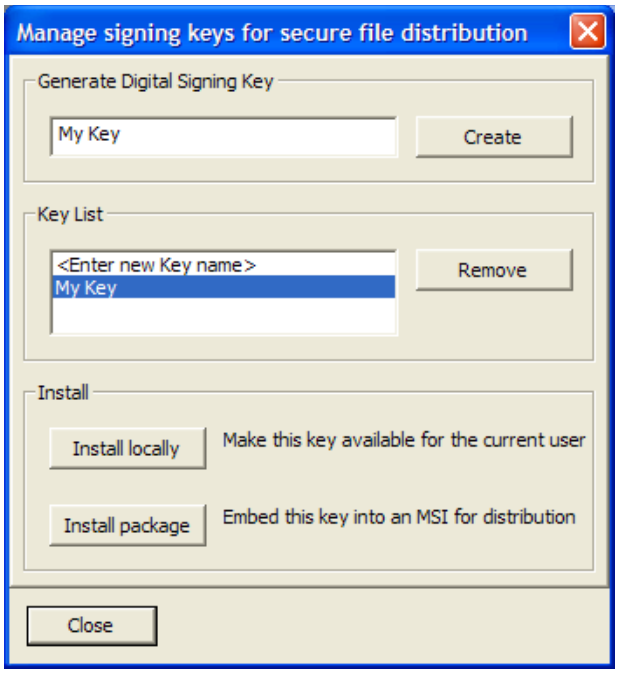
However, for example, if a user has configured a Hotmail application definition locally, and a predefined application is not supplied in the `.msi` file, the user's Hotmail application definition is not changed.

---

8 Click `Manage Keys`. The Manage signing keys for secure file distribution dialog box is displayed.

9 Specify a name for the key in the *Generate Digital Signing Key* field.

10 Click *Create*.



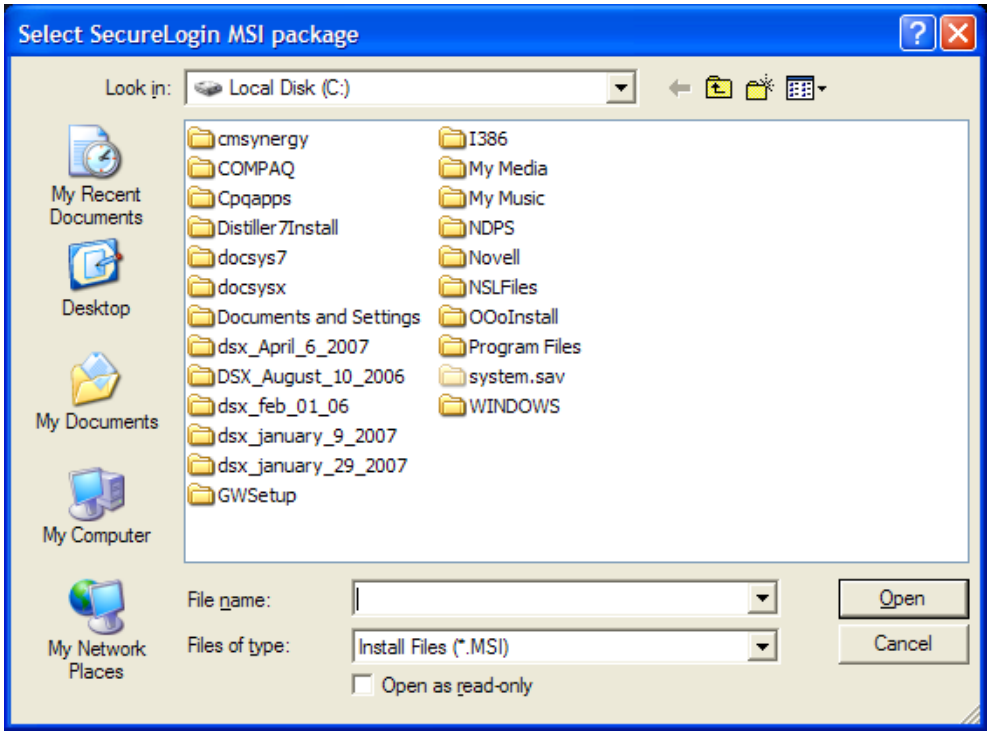
11 From the *Key List*, select the newly created key.

12 Under *Install*, click *Install Package*. The Load Settings dialog box is displayed.

13 Browse to locate the distribution file (`.msi` file) in which you want to embed the key.

14 Click *Open*. A confirmation message that the key is embedded in the `.msi` file is displayed.

15 Click *OK*.



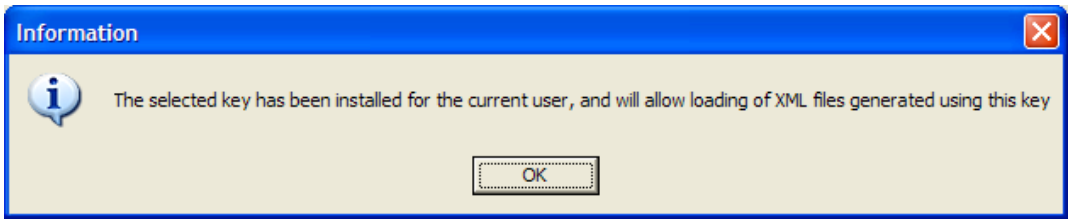
You can now distribute and install the `.msi` file on the user's machine. This allows them to import signs that are signed and encrypted.

After the keys are created, they must not be deleted because they are randomly generated. They key used must correspond to the key that is been previously packaged and with the distributed installer.

14.6 Locally Installing a Digital Signing Key

The *Manage signing keys for secure file distribution* dialog box provides a tool to install a digital signing key locally, enabling loading of XML files generated using this key

1. Log in to iManager.
2. Select *Securelogin SSO* > *Manage Securelogin SSO*. The Manage SecureLogin SSO page is displayed.
3. In the object field, specify your object name, then click *OK*.
4. Click *Distribution*. The Distribution details are displayed.
5. Click *Manage Keys*. The Manage signing keys for secure file distribution dialog box is displayed.
6. Specify a name in the *Generate Digital Signing Key* field.
7. Click *Create*.
8. From the *Key List*, select a new key.
9. Under *Install*, select *Install Locally*. A confirmation message appears.



10. Click *OK*.

Using The SLAP Tool

15

This section provides information on the following:

15.1 About The SLAP Tool

The SecureLogin Attribute Provisioning (SLAP) tool uses command line options to allow SecureLogin to leverage user data from an organization’s provisioning system. Using the SLAP tool, you can import data, in XML format from third-party applications into the SecureLogin user’s datastore as well as export information (except user application passwords and the user’s passphrases).

Data that can be manipulated includes:

- User variables
- Application definitions
- Organizational settings
- Password policies
- Logins
- Passphrase questions and answers

The SLAP tool command operates as a provisioning tool between SecureLogin data in a directory and in an XML file. The XML schema used is the same as the Copy Settings GUI importer/exporter. In addition to copying settings, the SLAP tool can extract usernames. The SLAP tool cannot export single sign-on sensitive data such as passwords and passphrases.

For example, an organization with 10,000 users in a SAP* system, implementing SecureLogin can speed deployment significantly by automating the initial user login. To do this, use a file containing multiple users’ username and password combinations from SAP, and use the SLAP tool to import the file into the SecureLogin datastore as a bulk process. The SLAP tool removes the requirement for each user to enter credentials on the first log in to SecureLogin.

NOTE: When the SLAP tool is used for initial provisioning of SecureLogin user accounts, before any SecureLogin data has been stored for users, the XML file must include a passphrase question and response. This question/response can be the same for each user and can be changed by the user after deployment.

15.2 The SLAP Syntax

```
slaptool [-hlaspcPef] -r object_name_file | -o "object" [file ...]
```

The following table describes the command options.

Command	Description
-h	Displays a help message and exits (all other options are ignored).
-l	Excludes user IDs.
-v	Excludes variables (passwords will not be exported in the current version).
-a	Excludes applications.
-s	Excludes settings.
-p	Excludes password policies.
-c	Excludes credsets.
v	Excludes passphrases (affects an import only).
-e	Performs an export rather than an import.
-r	object_name_file Specifies a file containing line-delimited object names on which to perform the operation.
-o	object Specifies a particular object on which to operate.
-f	Uses the cache file, rather than accessing a directory. Cannot be used with -r or -o, and SecureLogin must be set to use Dummy mode. The user is selected interactively at run time).
[file]	Specifies one or more .XML files from which to read data (or to write to for exporting). No file specification. It reads and writes data from and to the stdin and stdout. For example: `./slaptool.exe -o "CN=bernie.O=activcard.T=DEVTEST" initial_setup.xml` This reads userIDs, applications, settings and password policies from the file initial_setup.xml and writes them out to the object: "CN=bernie.O=activcard.T=DEVTEST"
-k [password]	Enables the creation of a passphrase answer for individual users in LDAP and Microsoft Active Directory environments. It is mandatory for users to save a passphrase answer on first log in to SecureLogin. The SLAP tool requires password authorization to save user data. The -k switch provides the user password, enabling automated creation of the passphrase answer. This answer can be manually changed by users after provisioning. For example, the following command is used to import user data and a passphrase question and answer combination: `slaptool.exe -k password -o context filename.xml` This reads userIDs, applications, settings, and password policies from the file initial_setup.xml file and writes them out to the object: "CN=writer.O=novell.T=DEVTEST"

SLAP Tool Example

The following Perl application definition, created for the example organization discussed previously, assumes that usernames and passwords are stored in a text file named `listofnames.txt`. There is one space between each username and password pair per line.

A XML file, such as the "XML File Example" on page 157 is required to run this application definition, containing the data for import. Where the data is customized on a per user name basis, the string to be substituted is replaced with *usernamegoeshere*.

For example:

```
**************************************************
open FILE,"listofnames.txt";
foreach (<FILE>) {
chomp;                  # Clean string
@lines = split(/\n/);   # Split up string
for each $l (@lines) {
    @fields = split(/\s/);
    $name = $fields[0];
    $pass = $fields[1];
    open DATAFILE,"source.xml";
    open OUTFILE,">data.xml";
    foreach (<DATAFILE>) { # Write up a file specific to this user
        s/\*usernamegoeshere\*/$name/;
        s/\*passwordgoeshere\*/$pass/;
        # Any other variable substitution can be done here too...
        print OUTFILE "$_";
   }
    close DATAFILE;
    close OUTFILE;
   system "slaptool.exe -k \"$pass\" -o
\"CN=$name.O=myorg.T=OURCOMPANY\" data.xml";
        }
}
close FILE;
unlink 'data.xml';
**************************************************
```

Using an XML file called `source.xml`, run the application definition with the data that is to be imported. For example, you can manually export data from a single user setup with the value for the username replaced with the string "*usernamegoeshere*".

NOTE: The example application definition does not include error handling.

XML File Example

```
<?xml version="1.0"?>
<SecureLogin>
   <passphrasequestions>
      <question>Please enter a passphrase for SLAP testing.</question>
   </passphrasequestions>
   <passphrase>
      <activequestion>Please enter a passphrase for SLAP
testing.</activequestion>
      <answer>passphrase</answer>
   </passphrase>
```

```
   <logins>
      <login>
         <name>fnord</name>
         <symbol>
            <name>username</name>
            <value>bob</value>
         </symbol>
         <symbol>
            <name>Password</name>
            <value>test</value>
         </symbol>
      </login>
<login>
         <name>notepad.exe</name>
         <symbol>
            <name>username</name>
            <value>asdf</value>
         </symbol>
         <symbol>
            <name>Password</name>
            <value>test</value>
         </symbol>
      </login>
      <login>
         <name>testlogin</name>
         <symbol>
            <name>username</name>
            <value>Novell</value>
         </symbol>
         <symbol>
            <name>Password</name>
            <value>test</value>
         </symbol>
      </login>
    </logins>
</SecureLogin>
```

Managing the Workstation Cache 16

This section provides information on the following:

16.1 About the Workstation Cache

The SecureLogin cache is an encrypted local copy of SecureLogin data. It allows users who are not connected to the network (or working offline using a laptop) to continue to use SecureLogin even if the directory becomes unavailable.

User data includes credentials, preferences, policies, and SecureLogin application definitions, except when you use a smart card for storing credentials. By default, a cache file is created on the workstation as part of SecureLogin installation. The cache file stores user data locally and is synchronized regularly with the user’s data in the directory. You can set this in the Administrative Management utility. You can also disable cache synchronization, storing all user data in the directory.

Depending on the type of installation, the cache can be stored under <Path to SecureLogin >\Cache. For example: `C:\Program Files\SecureLogin\Cache`

The same can be stored in the user's profile, for example, `C:\Users\<Username>\Application Data\SecureLogin\Cache` on Windows Vista

Directory and workstation caches are synchronized regularly, by default every five minutes, and whenever the user logs off or on to the workstation. When changes are made, either by the user on the workstation or the administrator in the directory, single sign-on user data is compared and updated during synchronization. Any settings configured by the user through the Credentials Management tool on the local workstation take precedence over those made in the directory.

If you require full administrative control of a user’s SecureLogin environment, you can disable the user's access to administration tools through the settings in the Preferences Properties table. This prohibits users from overriding your changes while configuring changes on the workstation.

NOTE: The SecureLogin cache refresh interval is by default five minutes. You can change the default in the Preferences Properties table.

Because SecureLogin data is stored in the directory, existing directory backups also back up SecureLogin data. In addition, the local cache synchronizes with the directory for further redundancy of data. Backing up or restoring by using the SecureLogin menu options is typically performed by users who have been disconnected from the network for long periods of time, such as weeks or months.

Using workstation backup and restore, users can securely back up their SecureLogin cache in stand-alone or directory deployments. All user data, including passwords and passphrases, is saved in a password-protected, encrypted XML file.

16.2 Creating a Backup File

1. In the notification area, right-click the Novell SecureLogin icon, then select *Advanced* > *Backup User Information*. The Save Settings dialog box is displayed.
2. Select a folder to store the backup file. The file can be stored in any location.
3. In the *File name* field, specify a name for the backup file.
4. Click *Save.* The Password dialog box is displayed.
5. In the *Password* field, specify a password.
6. Click *OK*.

   The encrypted and password-protected backup file is saved, and a confirmation message appears.
7. Click *OK*.

16.3 Deleting the Workstation Cache

IMPORTANT: Before restoring the backup file, you must delete the cache file on the workstation. In directory environments, you must also delete the user object data in the directory.

1. Right-click the Windows *Start* button, then click *Explore*.
2. Browse to the following directory: *C:\Documents and Settings\[user]\Application Data\SecureLogin\Cache*

   Ensure that you have selected `Show hidden files and folders` in the Windows Folder Options dialog box.
3. Delete the cache directory.
4. Close Windows Explorer.

16.4 Restoring the Local Cache Backup File

1. In the notification area, right-click the Novell SecureLogin icon, then select *Advanced* > *Restore User Information*. The Load Settings dialog box is displayed.
2. Select the backup file.
3. Click *Open*. The Password dialog box is displayed.
4. In the *Password* field, specify the password.
5. Click *OK*.

   A message appears, confirming that cache data has been loaded to the local workstation cache.
6. Click *OK*.

Auditing

17

This section contains the following information:

17.1 About Auditing Tools

SecureLogin provides monitoring functionality with Simple Network Management Protocol (SNMP) trapping and Windows event logging. SecureLogin’s support for both of these auditing tools allows you to choose a preferred auditing application and to integrate event monitoring into your current SNMP functionality. Event alerts are activated through SecureLogin application definitions. An understanding of application definition is useful to enable event monitoring.

17.2 Sending SNMP Alerts

You can send SNMP alerts from a client workstation to a specified console. This requires an SNMP console application on the receiving console, and the following SecureLogin files:

- `slsnmp.exe`
- `libsnmp.dll`
- `SecureLogin.mib`

The `slsnmp.exe` and `libsnmp.dll` files are provided in the `Tools` folder of the SecureLogin installer package. Copy the files to the following location on the client workstation:

```
<local drive>\Program Files\novell\SecureLogin\
```

The `SecureLogin.mib` file is imported to the SNMP trap console to decode the SNMP traps sent by SecureLogin.

Alerts are enabled in the SecureLogin application definition for the application. Through the SecureLogin application definition `Run` command, the alert is sent to the specified workstation IP address as well as the SNMP application active on this computer.

17.3 Scripting for SNMP Auditing

The following examples use the Windows Notepad application. Although Notepad does not require you to log in, you can create an application definition to respond to the execution of almost any application and to elicit additional information, such as the machine name, as a SNMP alert.

17.3.1 Prerequisites

- Identify the IP address of the receiving computer.
- Ensure that the SNMP console is active.

1. Close the Personal Management utility if it is open.
2. Start Notepad.
3. In the notification area, right-click the Novell SecureLogin  icon, then click *Add Application*. The Add Application Wizard is displayed.
4. Follow the prompts to enable the application.
5. In the notification area, double-click the Novell SecureLogin  icon to open the Personal Management utility.
6. Click *Applications*.
7. Double-click the application description. In this example, it is Untitled - Notepad.The Application pane is displayed.
8. Click the *Definition* tab. The application definition editor is displayed.

   The following example command sends an SNMP alert to the computer running the SNMP console application, advising that Notepad has been activated.

   You can set alerts for any event that SecureLogin responds to, including Change Password dialog boxes and error messages.
9. After the EndDialog command, type the following:

   ```
   Run "C:\Program Files\novell\SecureLogin\slsnmp.exe" public <IP address>
   "Notepad has started"
   ```
10. Click *OK* to save the command and to close the Personal Management utility.
11. Start Notepad. The alert is sent to the SNMP console.

17.4 About Windows Event Log Alerts

Windows event log alerts are activated by following the same procedure as SNMP alerts. The `Logevent.exe` application is activated through the `Run` command in an application definition.

Windows event logging from SecureLogin requires that the Windows Event Log system is active on the computer receiving the alerts, along with the executable `Logevent.exe` on each audited client workstation, to generate the alerts.

NOTE: `Logevent.exe` is included in the Windows 2000 Resource Kit.

17.5 Creating a Windows Event Log Alert

The following procedure uses the Windows Notepad application as an example.

1. In the notification area, double-click  to open the Personal Management utility.
2. Click *Applications*.
3. In the right pane, double-click the application description (in this example, Untitled-Notepad). The Application Pane is displayed.

4 Click the *Definition* tab. The application definition editor is displayed.

5 The command syntax to execute `LogEvent.exe` is:

```
logevent -m \\computername-s severity-c categorynumber-r source-e eventID-
t timeout"event text"
```

Definitions of the command parameters and event IDs are also available on the Microsoft Web site.

6 After `EndDialog`, specify the `LogEvent` command for the required alert.

For example:

```
Run "C:\Program Files\Resource Kit\LogEvent.exe –m SecureLogin –s –e
99"Notepad has started"
```

This command requests an alert to be sent to the console with a security level of W – warning and event ID number 99.

7 Click *OK*.

8 Start Notepad. The alert is sent to the Windows Event Log system.

Novell Audit Configuration for Novell SecureLogin

18

Novell Audit has two primary components, the Secure Logging Server and the Platform Agent. The Secure Logging server receives and processes events from all other services on the network. The Platform Agent runs on all the SecureLogin workstations that you want to audit.

To configure Novell Audit, perform the following tasks:

18.1 Installing the Platform Agent

To install Novell Platform Agent on Windows:

1. Double-click `naudit_win32.exe`, from the Novell Audit installer package. The Install Type dialog box is displayed.
2. Select *Instrumentation*, then click *OK*. The eDirectory instrumentation and platform agent files are installed.
3. During the installation, you might be prompted to specify the logging server address. Specify the IP address of the SLS. This automatically updates the `C:\WINNT\logevent.cfg` with the correct LogHost entry.
4. (Optional) Edit `c:\winnt\logevent.cfg` and add an entry for `LogReconnectInterval`.

   ```
   LogReconnectInterval=60
   ```

For detailed information on installing the Platform Agents see, the Novell Audit Documentation Web site. (http://www.novell.com/documentation/novellaudit20/index.html?page=/documentation/novellaudit20/install/data/bux9mi0.html)

18.2 Pointing Platform Agents to the Logging Server

You can point the platform agents to the Secure Logging Server during the platform agent installation, or you can modify the platform agent configuration file, `logevent.cfg`, to reflect the location. This file is available in the Windows directory if the Platform Agent is installed (`Winnt` for Windows 2000, `Windows` for Windows XP.

18.3 Configuring the Secure Logging Server Using iManager

If you use iManager on a Novell Open Enterprise Suite (OES) server, the Audit plug-in for iManager is already installed. Otherwise, download and install the Novell Audit plug-in from the Novell Web site (http://download.novell.com/).

This section contains the following information:

18.3.1 Logging Events to the Appropriate Channel

1. Log in to iManager.
2. Select *Auditing and Logging* > *Logging Server Options*.
3. Browse and select the logging server installed in the tree. It is typically located under *Root* > *Logging Services* > *Server_Name* > *Logging Server*.
4. Click *General*.
5. In the *Log Channel* field under the *Configuration* section, browse and select the required channel. For example:

   For files: `File.Channels.Logging Services`

   For MySQL: `MySQL. Channels.Logging Services`
6. Click *Channels*.
7. Select the required channel and edit the channel information to provide information about where the events are logged.
8. Click *Apply*.

18.3.2 Reconfiguring Secure Logging Server with the SecureLogin Audit Schema

1. Click *Log Applications*.
2. Select the *Applications* check box.
3. Select *New Log Application*.
4. Type *SecureLogin* in the *Application* field.
5. Browse to the `SecureLogin.lsc` file available in `SecureLogin\Tools` directory in the SecureLogin installer package.
6. Click *OK*.
7. On the *General* tab, select *Summary* and verify all the configuration settings.
8. Click *Apply*.

18.3.3 Setting SecureLogin Preferences

To enable logging from SecureLogin, set the following preferences:

1. Access the Administrative Management utility.

   For more information on how to access the Administrative Management utility see Section 1.2, "Starting the Administrative Management Utilities," on page 14 and Section 1.3, "Accessing the Single Sign-On Plug-In Through iManager," on page 15.

2. Click *Preferences*.

3. In *General* Preferences, set the value of *Enable Logging to Novell Audit* to *Yes*.

4. Click *Apply*.

   The following events are logged:

```
Event ID 00330001: SSO AuditEvent Script Command
Event ID 00330002: SSO Client Started
Event ID 00330003: SSO Client Exited
Event ID 00330004: SSO Client Activated By User
Event ID 00330005: SSO Client Deactivated By User
Event ID 00330006: Password Provided By A Script
Event ID 00330007: Password Changed by the user in response to a
ChagePassword command
Event ID 00330008: Password Changed automatically in response to a
ChagePassword command
```

18.4 Configuring the Registry to Enable Logging from LDAP and the Secure Workstation

To log events from SecureLogin LDAP authentication module:

1. Enter `LdapAudit` as a registry value at:
   `HKEY_LOCAL_MACHINE\Software\Novell\Login\Ldap`

   The following events are logged:

```
Event ID00330021: NSL user login
Event ID00330022: LDAP user password change
Event ID00330023: Workstation unlocked by different User
```

To log events from Secure Workstation:

1. Enter `SWAudit` as a registry value at: `HKEY_LOCAL_MACHINE\Software\Novell\NMAS\MethodData\Secure Workstation`

   Following events are logged:

```
Event ID00330041: Inactivity Timeout
Event ID00330042: Device Removal
Event ID00330044: Manual Lock event
```

Administering Secure Workstation 19

This section contains information on the following:

19.1 Understanding Secure Workstation Policies

The Secure Workstation policy specifies how Secure Workstation behaves.

- The Local policy
- The Network policy
- The Effective policy

The Local policy is stored under an ACL-protected registry key on the workstation.

The Network policy is stored in eDirectory and delivered to the workstation using the NMAS Post-Login Method. For more information, see “Installing the NMAS Server Methods” in the *Novell SecureLogin 6.1 SP1 Administration Guide.*

The Effective policy is created by combining the Local policy with the Network policy.

Secure Workstation always enforces the Effective policy.

Secure Workstation reads the Local policy each time a user logs in to Windows. As long as the Novell Secure Workstation Service is running, the Local policy will be in effect during each user's Windows session.

When a user logs in to the network using the Secure Workstation Post-Login Method for NMAS, the post-login method sends the Network policy to the Novell Secure Workstation Service. The service reads the Local policy and combines it with the Network policy to create the Effective policy. The Effective policy consists of the most secure settings from the Local policy and the Network policy.

To see details about the policy that Secure Workstation is currently enforcing, click *View Effective Policy*, in the Secure Workstation’s main dialog box. If you have already started the Novell Secure Workstation service, it might not have an effective policy yet. If so, you get an error message when you click *View Effective Policy*. The service creates an Effective policy only when the user logs in to Windows, or when a user logs in using the Post-Login Method for NMAS.

The priority between the Local Policy and Network Policy depends on the action executed in the policy. For example, if the Local Policy is set to *Close All Programs* and the Network Policy is set to *Log Out of the Network*, the Effective Policy considers both these events and enforces the most secure settings of the Local and Network policy.

If a user logs in to Windows but does not use the post-login method, the service creates the Effective policy by making a copy of the Local policy.

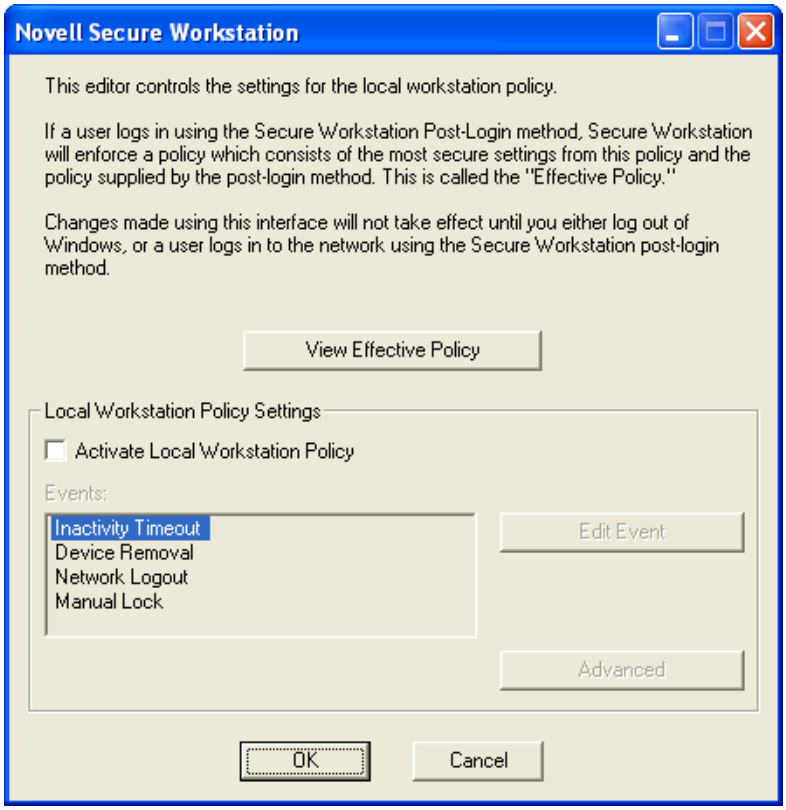
19.1.1 Setting the Secure Workstation Policies

When upgrading or uninstalling Novell SecureLogin, ensure that the Secure Workstation policies are configured in such a manner that they do not terminate the Novell SecureLogin installation.

19.2 Local Policy Editor

The Local Policy Editor provides an easy way to edit the Local policy. To access the Editor, click *Start* > *Programs* > *Novell SecureLogin* > *Secure Workstation Policy Editor*.

Figure 19-1 *Local Policy Editor*

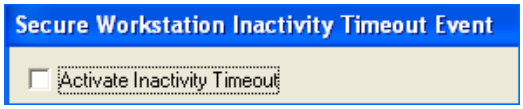


By default the Local policy is inactive, and most of the controls on the dialog box are inactive. To activate the Local policy (and all of the controls on the dialog box), select *Activate Local Workstation Policy*.

The Secure Workstation policy enables you to specify the lock events that Secure Workstation should watch for, and what action should be taken when an event occurs. The Events list box displays a list of lock events.

You can edit settings for a specific event by selecting the event in the list box and clicking *Edit Event*. A dialog box is displayed with settings for the event you select.

Figure 19-2 *The Edit Event Settings*

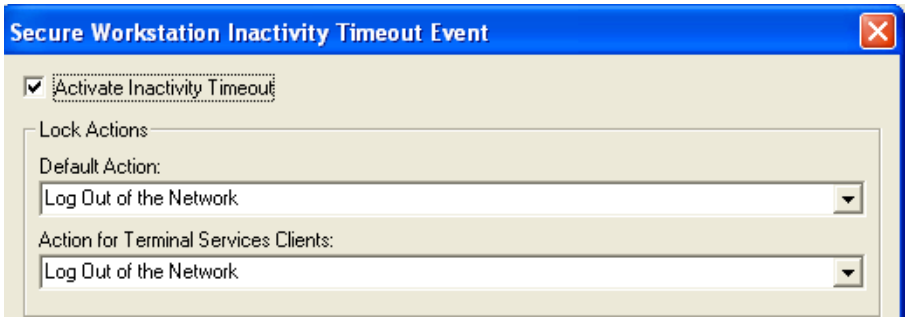


Secure Workstation ignores the event unless the Active check box is selected.

For each event, you can select from the following list of options:

- A drop-down list for selecting a default action
- A drop-down list for selecting an action for Terminal Services Clients.

Figure 19-3 *The Lock Actions*



The *Default Action* list contains the following items:

- *Log Out of the Workstation*

  Logs the user out of Windows.
- *Log Out of the Network*

  Logs the user out of either Client32™ or the LDAP Authentication Client, depending on which one has been installed.
- *Close All Programs*

  Closes a set of programs specified in the Advanced section of the policy.
- *Close All Programs and Log Out of the Network*

  Closes all the programs and logs out of the network.
- *Lock the Workstation*

  Causes the same result as pressing Ctrl+Alt+Del, then selecting *Lock Workstation*.

The *Action for Terminal Services Clients* list contains the following items

- *Log Out of the Workstation*

  Logs the user out of Windows.
- *Log Out of the Network*

  Logs the user out of either Client32™ or the LDAP Authentication Client, depending on which one has been installed.
- *Close All Programs*

Closes a set of programs specified in the Advanced section of the policy.

- *Close All Programs and Log Out of the Network*

  Closes all the programs and logs out of the network.

- *Disconnect the Session*

  Disconnects a remote terminal services session.

When a lock event is triggered, Secure Workstation performs the action associated with that event. Secure Workstation uses the default action on local workstations and the action for Terminal Services Clients for remote user sessions on remote workstations that are using either Citrix or Windows Terminal Services. Secure Workstation refers to these as “remote sessions”.

NOTE: If you are running the Local Policy Editor on a Terminal Server, the policy editor shows the Effective policy for the session that it is running in.

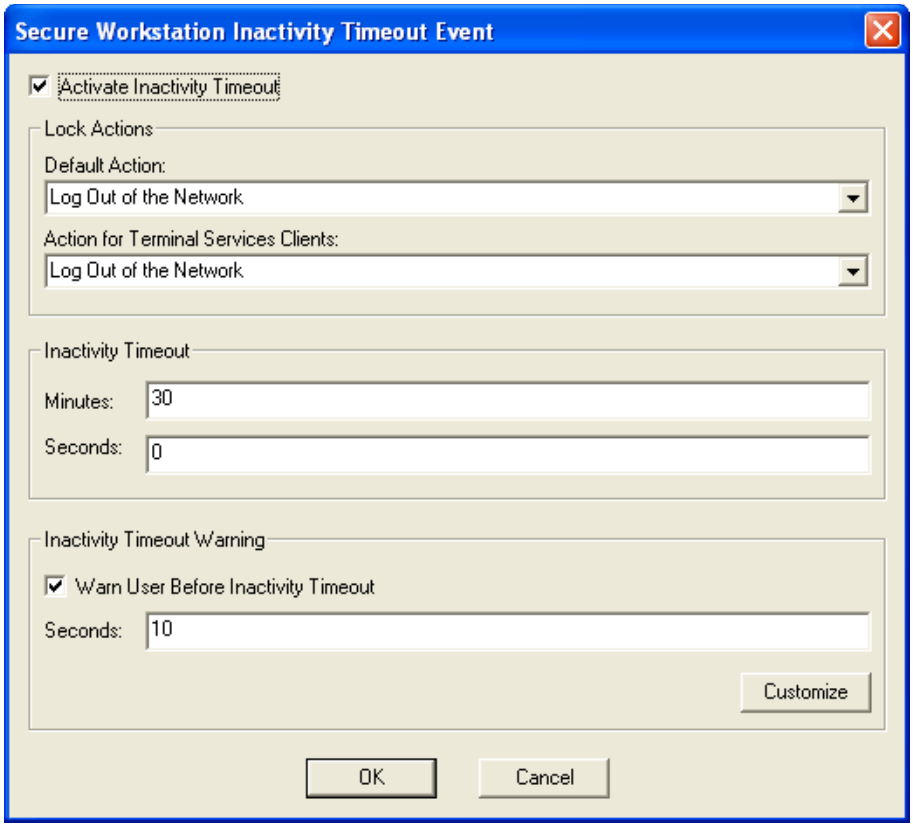
19.3 Configuring Secure Workstation Events

This section provides information on the following:

19.3.1 Configuring an Inactivity Timeout Event

The following figure illustrates the dialog box for configuring Inactivity Timeout events:

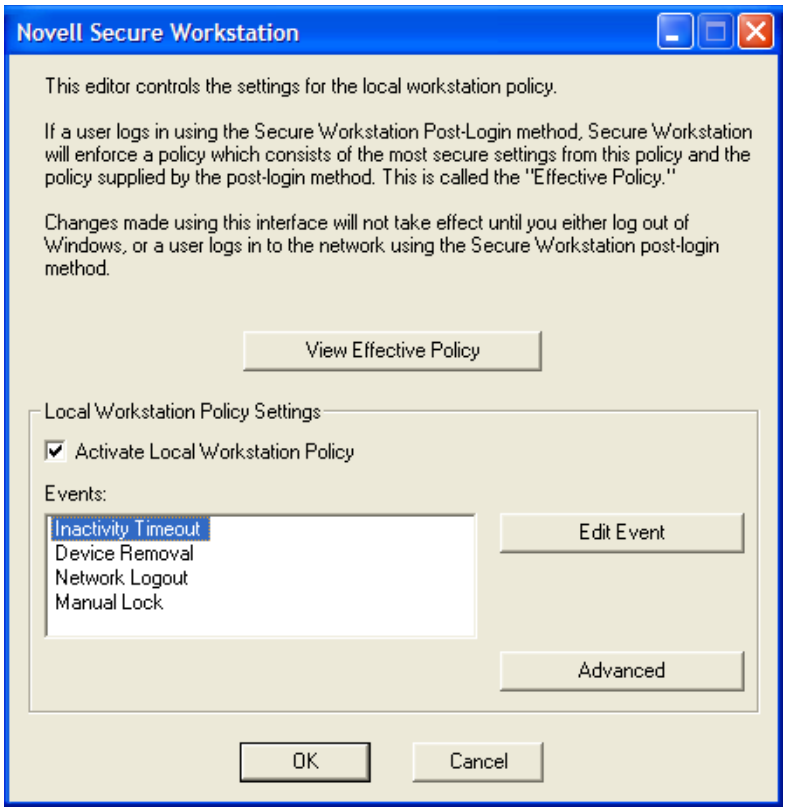
Figure 19-4 *Configuring Inactivity Timeout*



This dialog box enables you to specify the inactivity timeout and configure a warning that is displayed just before the inactivity timeout is reached.

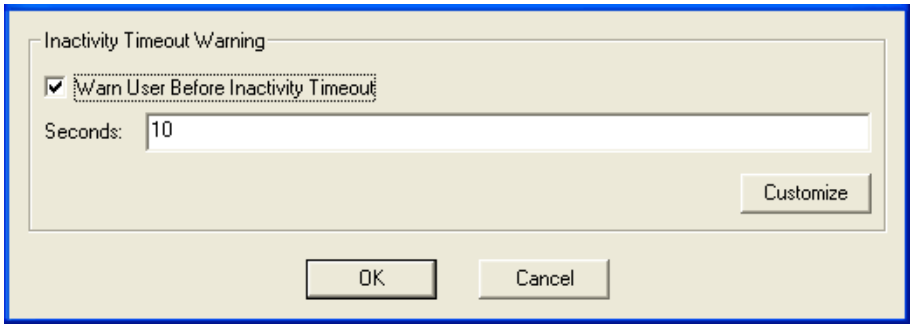
You can configure a `.wav` file to be played when the warning is shown. You can also specify an `.avi` file to be played for the warning. To configure these features:

1. Click *Start* > *Programs* > *Novell SecureLogin* > *Novell SecureWorkstation*. The local policy editor opens.
2. Under the *Events* list, click *Inactivity Timeout*.

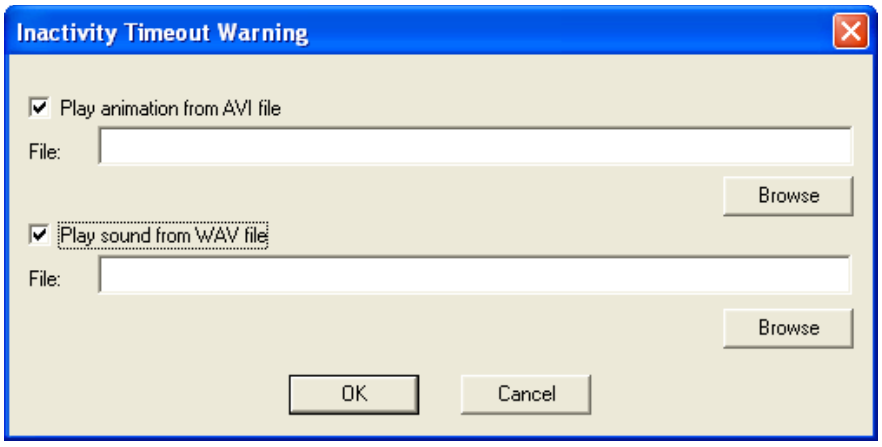


3 Click *Edit Event*.

4 Select *Warn User Before Inactivity Timeout* > *Customize*.



5 Select an option.



6 Browse to select `.avi` or `.wav` files.

7 Click *OK*.

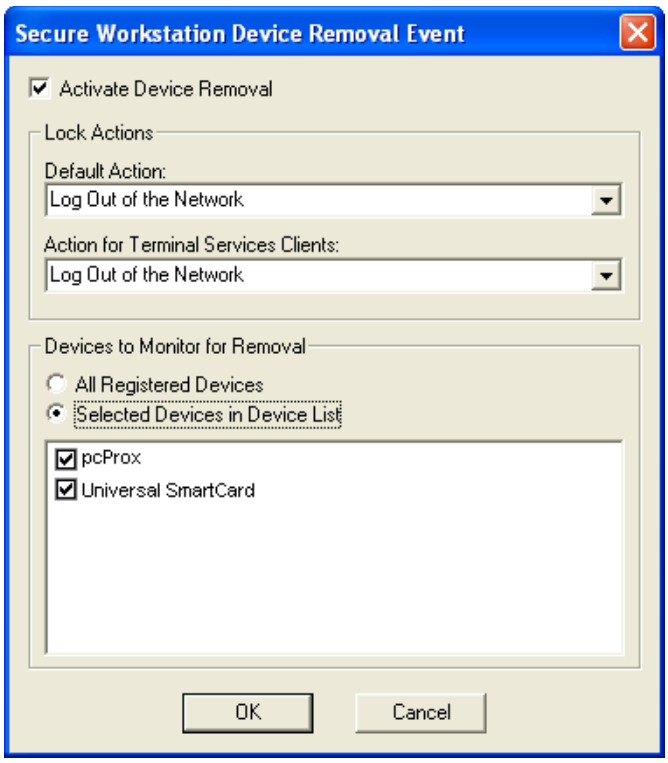
The warning message can accommodate `.avi` files that display images of any size.

The warning dialog box is displayed for the last few seconds of the inactivity timeout. You can specify the number of seconds that the warning dialog box is displayed. For example, if you set an inactivity timeout of thirty seconds and configure the warning dialog box to display for ten seconds, Secure Workstation displays the warning dialog box after twenty seconds of inactivity.

19.3.2 Configuring a Device Removal Event

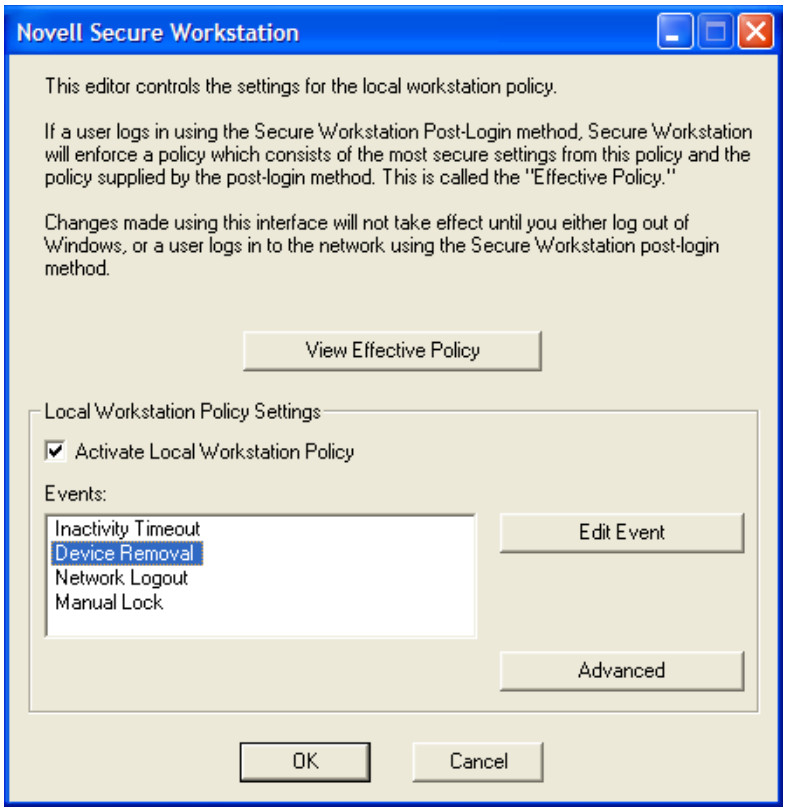
The following figure illustrates the dialog box for configuring a Device Removal event:

Figure 19-5 *Configuring a Device Removal Event*



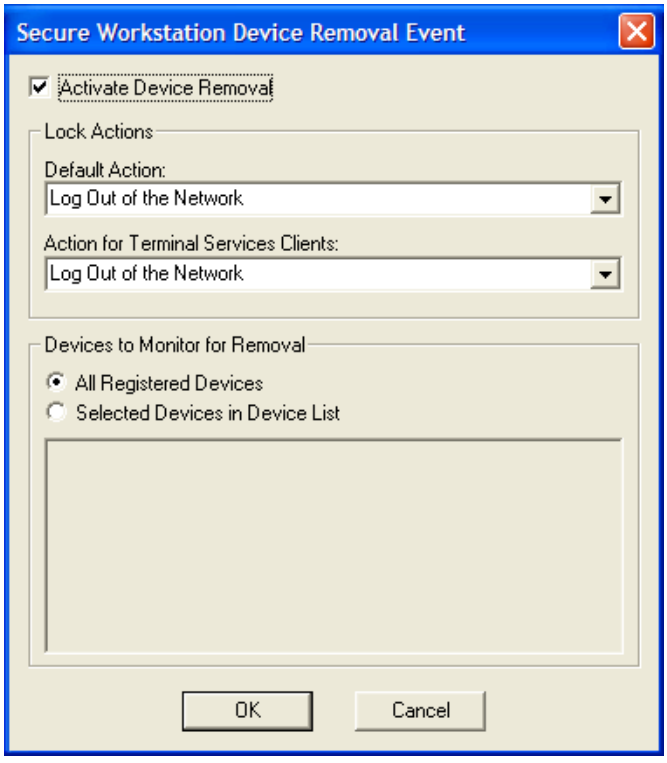
This dialog box enables you to specify which devices are included in the policy. If a device is included in the policy, it must be present during the user's session. If a device in the list is not present, Secure Workstation executes the lock action.

1. Click *Start* > *Programs* > *Novell SecureLogin* > *Novell SecureWorkstation*. The local policy editor opens.
2. Under the *Events* list, click *Device Removal*.



3 Click *Edit Event*.

4 Select *Activate Device Removal*.



5 Select the lock actions you want.

6 Select the devices to monitor:

- Select *All Registered Devices* if you want to monitor all the devices that are registered.
- Select *Selected Devices* in Device List if you want to monitor specific devices, then select the devices you want to monitor.

  The Devices to Monitor for Removal section contains a list of devices that are registered with the Secure Workstation.

  For Novell SecureLogin, both the Universal Smart Card and pcProx Methods for NMAS can report device removal events to Secure Workstation.

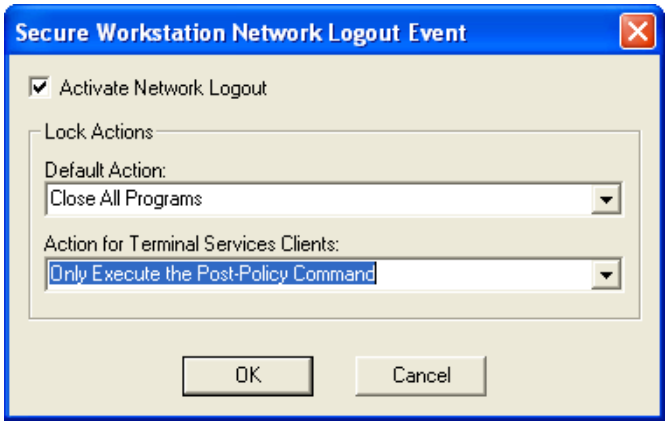
  Other NMAS partners have also implemented devices that can report device removal events to Secure Workstation. If you want to use a device that does not show up in the list, make sure that you have installed the NMAS Login Client Method for the device. If the device still doesn't show up, check with the vendor of the device to ensure that it will work with Secure Workstation.

7 Click *OK*.

19.3.3 Configuring a Network Logout Event

The following figure illustrates a Network Logout event:

Figure 19-6 *Configuring a Network Logout Event*



A Network Logout event is triggered when a user logs out of the network. This event could be triggered by either Client32 or the LDAP Authentication Client, depending on which client is present.

One of the intended uses of the Network Logout event is to close programs that the user might have used for single sign-on through Novell SecureLogin. This event might also be used to display a login dialog box or run a script when the user logs out. For more information, see "The Post-Policy Command" on page 182.

This event has a different set of lock actions than the other events. The *Default Action* list contains the following actions:

- *Log Out of the Workstation*

- *Close all programs*
- *Only Execute the Post-Policy Command*

The *Action for Terminal Services Clients* list contains the following actions:

- *Log Out of the Workstation*
- *Close All Programs*
- *Disconnect the Session*
- *Only Execute the Post-Policy Command*

The *Default Action* list does not include the following actions:

- *Lock the Workstation*

  This action has been omitted because of the behavior of the GINA. If a network connection isn't present when the workstation is locked, the Client32 GINA won't allow the workstation to be unlocked with an eDirectory authentication.

- *Log Out of the Network*

  This action has been omitted because it doesn't make sense to log out of the network in response to a network logout event.

The *Network Logout* event is the only event that includes the *Only Execute the Post-Policy Command* action. This action is actually a substitute for the *Log Out of the Network* action that is available with other events. If you want to execute a `Post-Policy` command on network logout, but not do anything else, use this action.

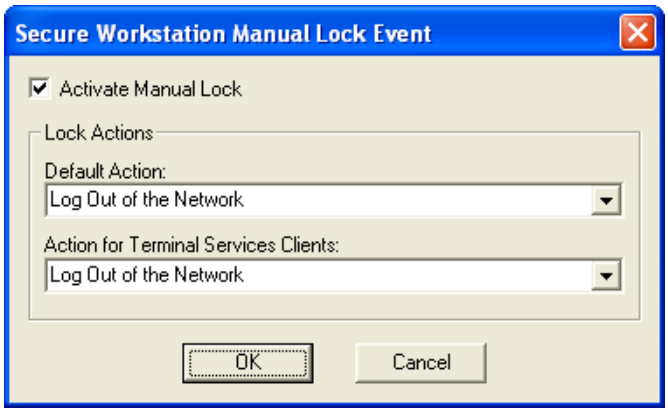
You can use the `Post-Policy` command to display a login dialog box or run a script. For more information, see "The Post-Policy Command" on page 182.

19.3.4 Configuring the Manual Lock Event

The *Manual Lock* event gives users the ability to manually trigger Secure Workstation. A user can manually trigger Secure Workstation either by clicking the *Logoff* button on the *Quick Logon/Logoff* Interface or by executing `SWLock.exe` in the `System32` directory.

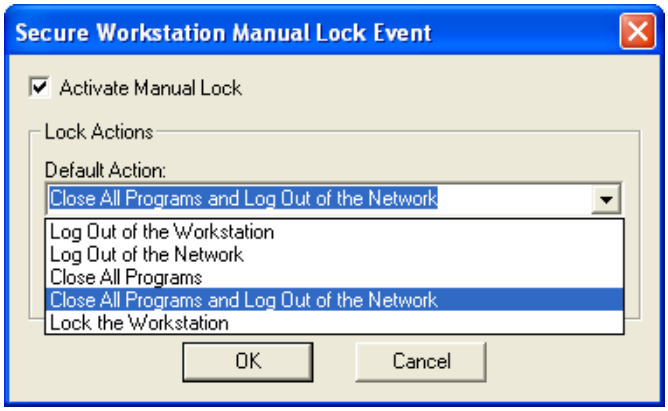
The following figure illustrates the *Manual Lock* dialog box:

Figure 19-7 *Configuring the Manual Local Event*



To configure Manual Lock:

1. Select *Manual Lock* from the main page, then click *Edit Event*.
2. Select the *Activate Manual Lock* check box.
3. (Optional) Select an option from the *Default Action* drop-down list.

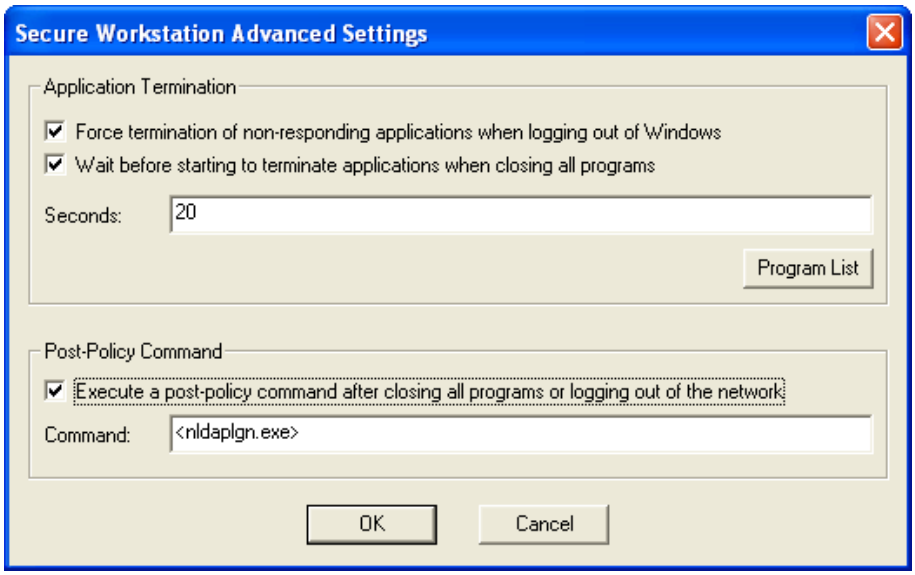


4. (Optional) Select an option from the *Action for Terminal Services Clients* drop-down list.

19.3.5 Advanced Settings

The following figure illustrates the Advanced Settings dialog box:

Figure 19-8 *The Advanced Settings*



To configure advanced settings, click *Advanced* on Secure Workstation's main dialog box.

Terminating Applications

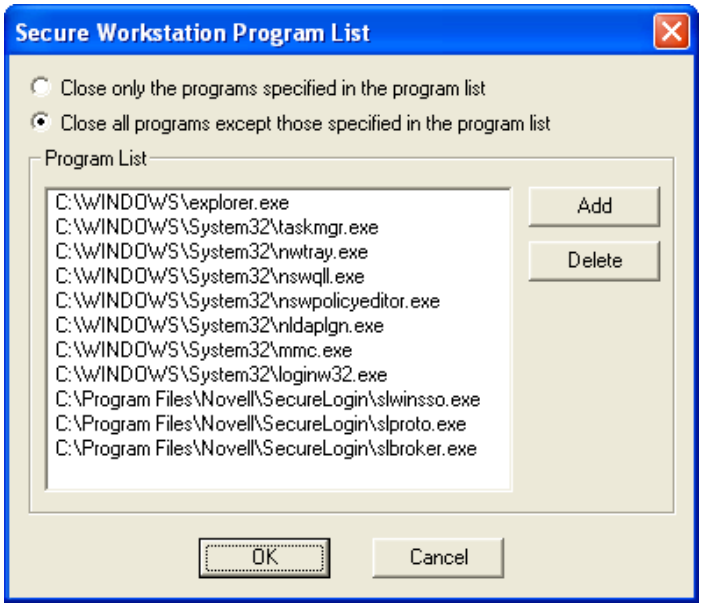
The Force Termination of Non-Responding Applications When Logging Out of Windows check box affects the way programs are shut down when Secure Workstation logs a user out of Windows. If this check box is selected, Windows terminates programs that do not respond to a Close message in a timely manner. This setting logs the user out of Windows more quickly, but some programs might not get an opportunity to save their data before being terminated.

The *Wait Before Starting to Terminate Applications When Closing All Programs* check box is similar, except that it controls the behavior of the *Close All Programs* action. When Secure Workstation closes programs, it always sends a Close message to each program to tell it to shut down. If the *Wait Before Starting to Terminate Applications When Closing All Programs* check box is not selected, Secure Workstation does nothing else to close the programs. The result is that some programs might not shut down.

For example, if Microsoft Word has an unsaved document, Secure Workstation might display a *Save As* dialog box.

On the other hand, if the W*ait Before Starting to Terminate Applications When Closing All Programs* check box is selected, Secure Workstation checks to see if the programs are still running after the specified timeout. Any programs that are still running at this point are terminated and might not have a chance to save their data.

You can use the *Program List* to specify which programs should be closed when Secure Workstation executes a *Close All Programs* action.



If you select *Close Only the Programs Specified in the Program List*, Secure Workstation closes only the programs listed.

If you select *Close All Programs Except Those Specified in the Program List*, Secure Workstation closes all programs except those specifically listed.

NOTE: If you select *Close All Programs Except Those Specified in the Program List*, SecureLogin closes every program in the user's sessions except those listed. This closing includes `explorer.exe`, the process associated with the user's desktop.

Secure Workstation closes only the programs that the currently logged in Windows user has sufficient rights to close on his or her own. Programs that the user does not have rights to (such as a service running as the LocalSystem account) are not closed.

When Secure Workstation is running on a Terminal Server, only the programs in the current user's session are closed. Programs running in other users' sessions aren't affected.

You don't need to specify the full path and name of each program in the program list. For example, instead of adding `c:\winnt\system32\notepad.exe` to the list, you could just add `Notepad.exe`.

However, if you do not specify the full path, the entry affects to all programs with that name, regardless of the path. For instance, listing Notepad.exe in the list without the path would match both `c:\winnt\system32\notepad.exe`, and `c:\documents and settings\user\notepad.exe`.

You can also use environment variables in the program list. For example, you could specify `%systemroot%\System32\notepad.exe` instead of `c:\winnt\system32\notepad.exe`.

The Post-Policy Command

The `Post-Policy` command is a command that is executed after Secure Workstation executes the lock action. This feature was designed to display a login dialog box after a Close All Programs or Log Out of the Network action has been executed. However, you can use this feature to run any program or script. You must provide the full path and name of the program to run.

To display the login dialog box, use `loginw32.exe` for Client32. Use `nldaplgn.exe` for the LDAP Authentication Client. One of the programs is located in the `system32` directory, depending on the mode of installation.

If you have configured the Network Logout event, Secure Workstation restarts the program specified in the `Post-Policy` command if it terminates before a user is logged in. This allows the login dialog box to be displayed again if a user clicks *Cancel*. For more information on configuring events for Secure Workstation, see Novell Technical Information Document 3407572 - Registry Keys and Values Used By Secure Workstations (http://www.novell.com/support/search.do?cmd=displayKC&docType=kc&externalId=3407572&sliceId=SAL_Public&dialogID=8134523&stateId=0%200%208138534)

19.4 Configuring the Network Policy

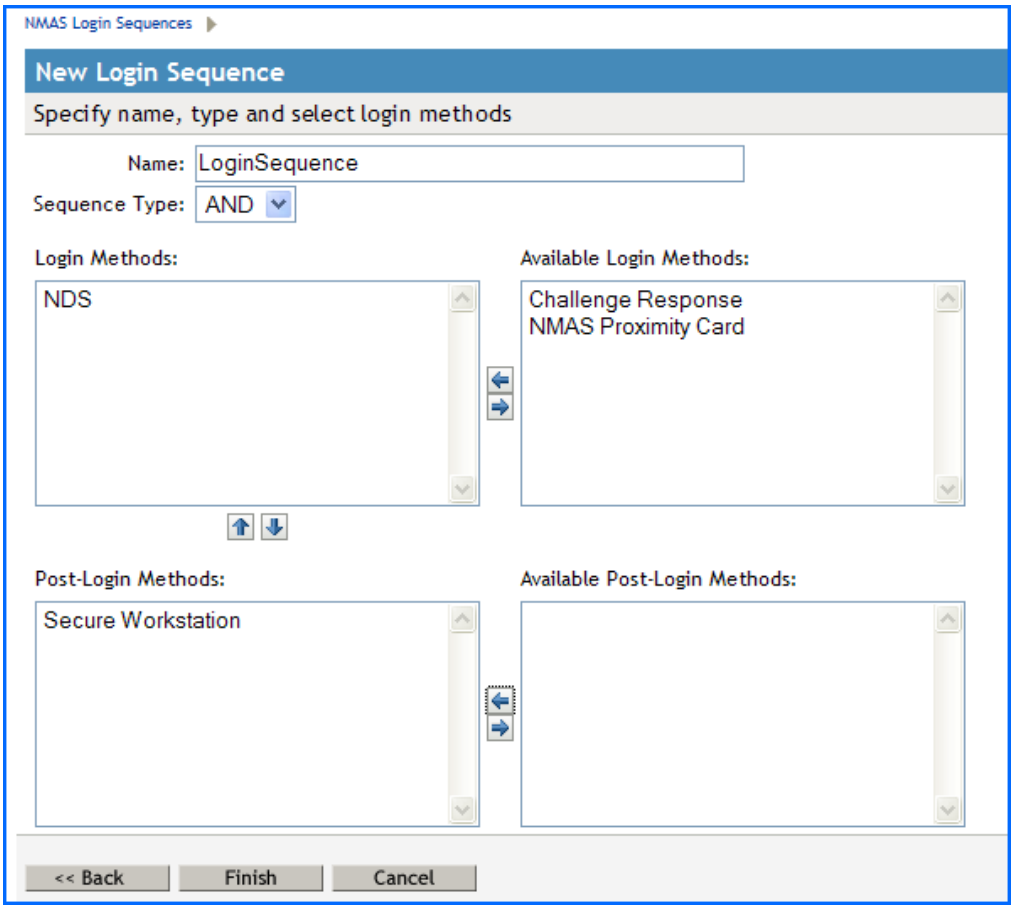
To configure a network policy:

1. Log in to iManager.
2. Select *NMAS* > *NMAS Login Sequences*. The NMAS Login Sequences page is displayed.
3. Click *New* and create a new login sequence.
   - **3a** Specify a name for the login sequence.
   - **3b** Select the sequence type from the *Sequence Type* drop-down list.
   - **3c** Specify the login methods.

The post-login method is Secure Workstation.

3d Click *Finish*.

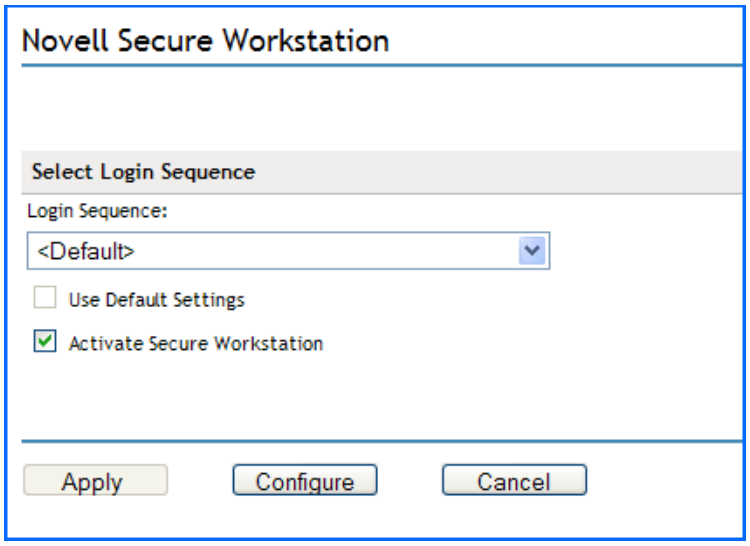
The login sequence is created successfully.



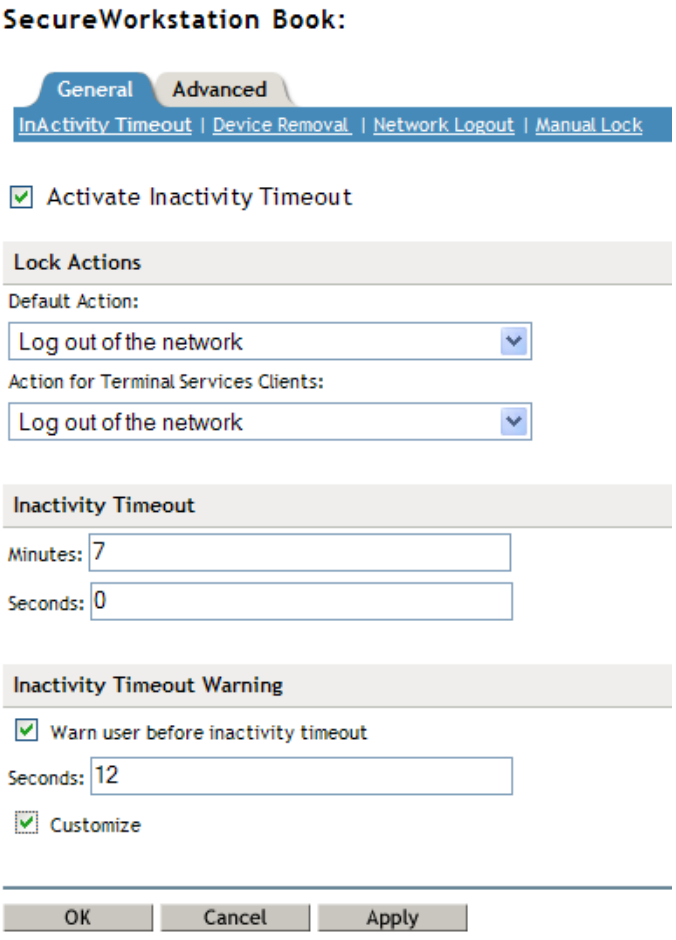
4 Select *Novell Secure Workstation* > *Select Sequence*. The Select Login Sequence page is displayed.

5 Select the login sequence you created.

6 Select *Activate Secure Workstation*.



7 Click *Configure*. The Secure Workstation page with the network policy configuring options is displayed.



Proceed to configure your network policy.

The procedures to configure the InActivity Timeout, Device Removal, Network Logout, Manual Lock, Application Termination, and Post-Policy Command are the same as explained in the following sections, earlier in this document.

Refer the following sections:

- Section 19.3.1, “Configuring an Inactivity Timeout Event,” on page 172
- Section 19.3.2, “Configuring a Device Removal Event,” on page 175
- Section 19.3.3, “Configuring a Network Logout Event,” on page 178
- Section 19.3.4, “Configuring the Manual Lock Event,” on page 179
- Section 19.3.5, “Advanced Settings,” on page 180

LDAP SSL Server Certificate Verification

20

This section contains the following information:

20.1 About LDAP SSL Server Certificate Verification

The LDAP SSL server certificate verification is a security feature that was introduced in the Novell® SecureLogin 6.0 SP1 release. This feature allows the client to verify the trustworthiness of the server, using a process similar to the certificate verification process carried out by browsers like Microsoft Internet Explorer and Mozilla Firefox. This certificate verification is similar to the certificate verification process carried out by browsers like Microsoft Internet Explorer and Mozilla Firefox.

Certificate verification of the server is important to prevent security hazards. It is essential that the client verify the server certificate during the LDAP SSL connection to the server. If the client cannot verify the server certificate, it is possible that an intruder on the same subnet can decrypt the communication between the client and access user credentials.

By default, eDirectory™ is configured with self-signed certificate. Although it works, it does not pass all the validation checks carried out during the verification process, so users are prompted whether to validate the certificate the first time they attempt to access the server. To prevent this, you can obtain a signed certificate from a known certificate authority such as VeriSign* and replace the existing certificate.

20.2 Validating an LDAP SSL Server Certificate

During LDAP connection, client receives the root certificate from the server so that client can verify the trustworthiness of the server. The client uses the following process to validate the certificate:

- It compares the current certificate with previously stored certificate, if any. If both certificates match, the client does not perform further checks, and adds the certificate to the local store. If the certificates do not match, the client continues the validation process.
- It checks whether the certificate is trusted. This ensures that a known authority is issuing the certificate.
- It checks whether the date on the certificate is valid with reference to the current date.
- It checks whether the host name on the certificate matches the date on the server.

If the certificate passes these preceding tests, the client adds the certificate to local store so it can be used for future verification.

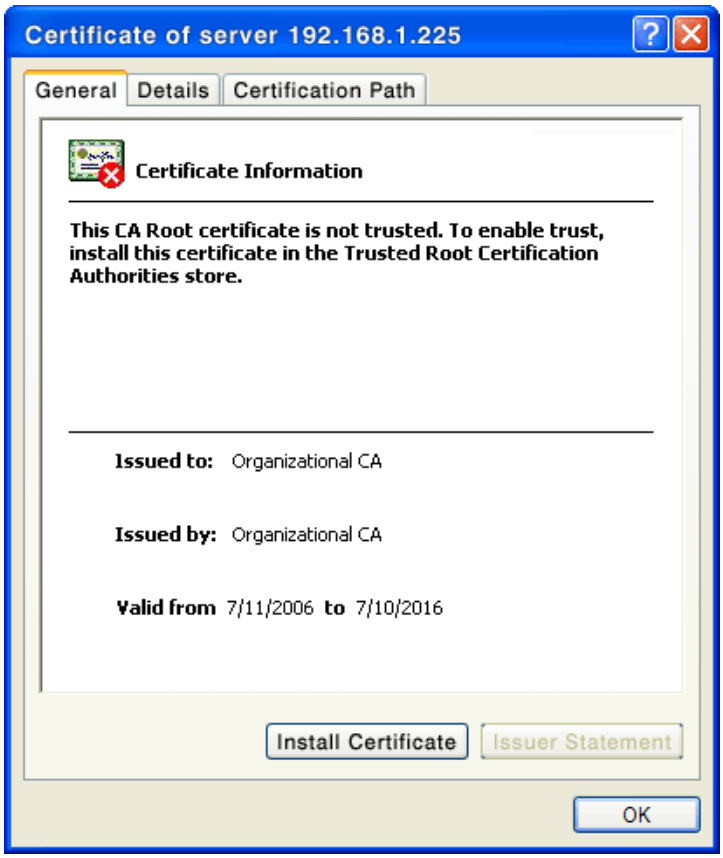
If the certificate does not pass the verification process, the application prompts you to either continue the connection or terminate the connection.

Figure 20-1 *Certificate Verification*

- To continue the connection, click *Yes*. The certificate is added to the local store so it can be used for future verification, and the authentication process continues.
- To terminate the connection, click *No*.
- To get details about the certificate, click *View Certificate* to display the Certificate Information dialog box shown in the above figure. If you decide that the certificate is valid, you can click *Install Certificate* to permanently install the certificate.

NOTE: This store is different from the local store used by LDAP client to store trusted root certificates.

Figure 20-2 *Certificate Information*



20.3 Enabling LDAP SSL Certificate Verification

By default, the certificate verification feature is disabled. You can enable this feature by adding the following registry value:

1. On the Windows *Start* menu, click *Start* > *Run* to display the Run dialog box.
2. Type `regedit` then click *OK* to open the Registry Editor.
3. Browse to the `HKEY_LOCAL_MACHINE\SOFTWARE\Novell\Login\LDAP` directory.
4. Create a DWORD Value file with the value 1. Name this file `VerifySSLCert`.
5. Exit the Registry Editor.

Novell SecureLogin Security Role Configuration for Active Directory

21

For a user to administer Novell SecureLogin in an Active Directory environment, a user must have both sufficient permissions to the Protocom attributes in the Directory that Novell SecureLogin utilizes for its credential store, as well as the correct Novell SecureLogin settings to allow the user access to specific Novell SecureLogin functionality.

The topics explained in this section are:

21.1 Directory Attributes

The protocom attributes hold user or container data that is used by Novell SecureLogin to provide Single Sign-On functionality. These attributes are named as follows:

protocom-SSO-Auth-Data
protocom-SSO-Entries
protocom-SSO-Entries-Checksum
protocom-SSO-Profile
protocom-SSO-Security-Prefs
protocom-SSO-Security-Prefs-Checksum

The function for each of these attributes is as follows:

protocom-SSO-Auth-Data:

- This attribute is only for a User object. It is an octet-string type.
- It contains all user-specific authentication data, such as the passphrase.

protocom-SSO-Entries:

- This attribute is for User, Container, and Organizational Unit objects. It is an octet-string type. This attribute contains the following:
- All the user's login user IDs and passwords
- Specific preferences and application definitions at the User object
- Corporate application definitions and preferences at the Container and Organizational Unit objects

protocom-SSO-Entries-Checksum:

- This attribute optimizes the loading of data from the Directory. Whenever data changes in the protocom-SSO-Entries attributes, the Checksum attribute is updated. When SecureLogin loads, it reads the checksum and compares it to the checksum in memory. If the checksums are different, SecureLogin reloads the Entries attribute from the directory.

protocom-SSO-Profile:

- This attribute is used to instruct SecureLogin to read the settings and preferences from another container.

protocom-SSO-Security-Prefs:

- This attribute stores data required for SecureLogin to operate before loading the users datastore. This data can include Administrator-set Passphrase questions, Passphrase help information, settings, and similar things.

protocom-SSO-Security-Prefs-Checksum:

- This attribute functions with the protocom-SSO-Security-Prefs attribute much like the protocom-SSO-Entries-Checksum functions with the protocom-SSO-Entries attribute.

21.2 Directory Permissions Assignment

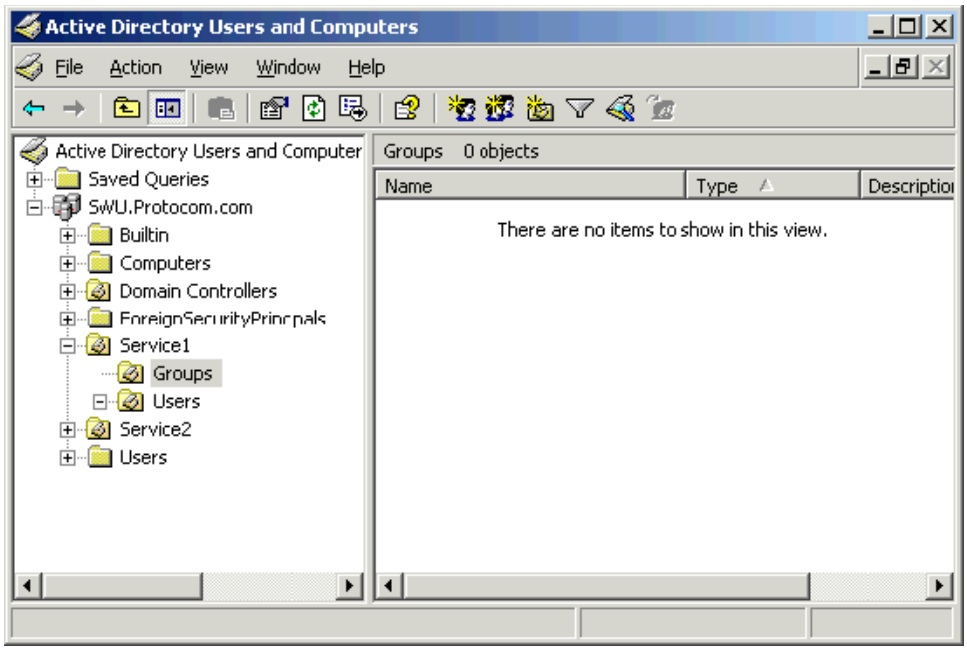
Based upon the above attribute descriptions and functions, specific roles might be granted the following permissions:

- Complete Novell SecureLogin Management:
  - protocom-SSO-Auth-Data = Read and Write
  - protocom-SSO-Entries = Read and Write
  - protocom-SSO-Entries-Checksum = Read and Write
  - protocom-SSO-Security-Prefs = Read and Write
  - protocom-SSO-Security-Prefs-Checksum = Read and Write
- Script, Credentials, and Clear Object Data administration:
  - protocom-SSO-Auth-Data = Read and Write
  - protocom-SSO-Entries = Read and Write
  - protocom-SSO-Entries-Checksum = Read and Write

Depending on the needs of your organization, these permissions can be assigned to specific users or groups at an organizational unit level. The following discussion demonstrates the creation of a SecureLogin Administration group and the delegation of permissions to an organizational unit that is one level below the top level organizational units in the Directory hierarchy.

21.3 Assigning Permissions for SecureLogin Administrators

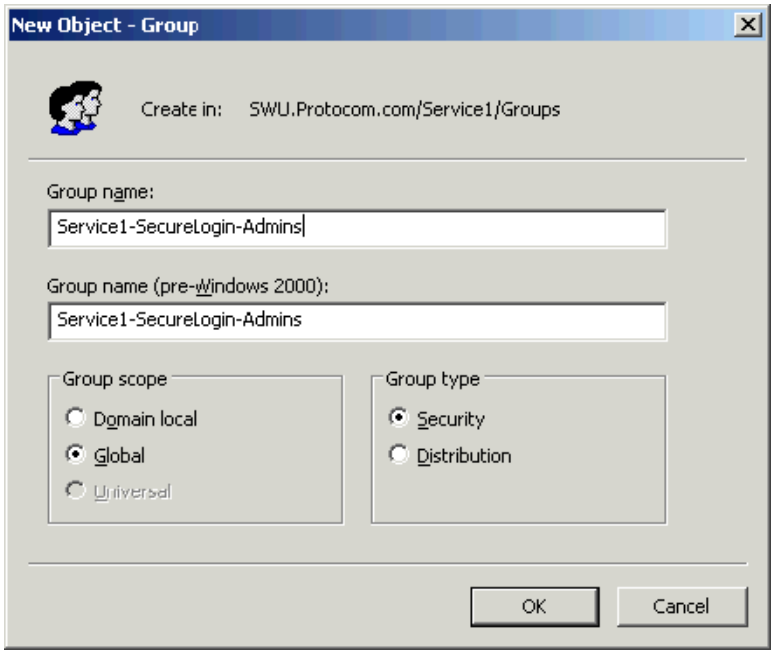
1. Login to the Active Directory domain as an administrative level user.



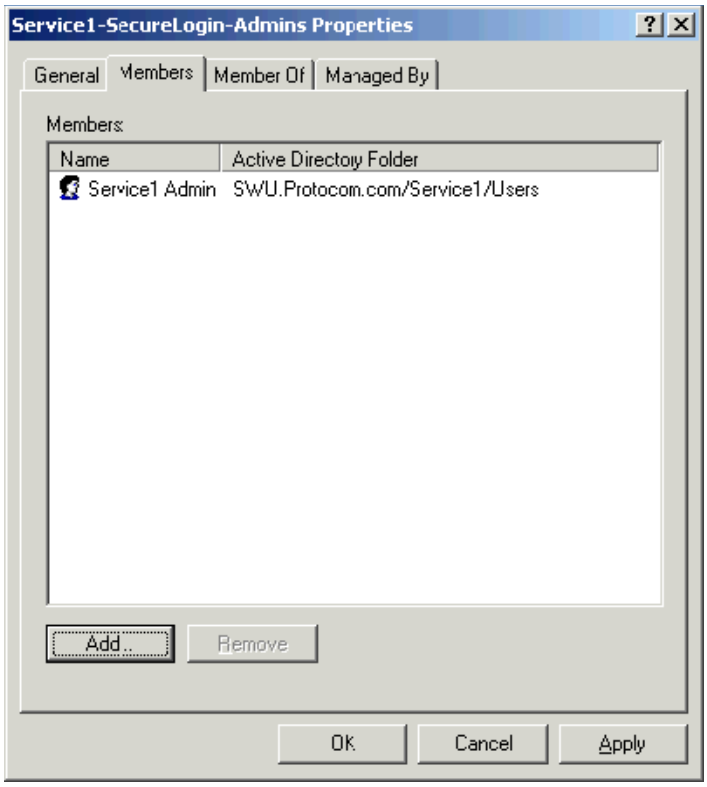
2 On a workstation or server, open *Active Directory User and Computers* (dsa.msc), and browse to the OU where you would like to create the group that will manage SecureLogin for the selected container and its children.

3 Click the create group button 

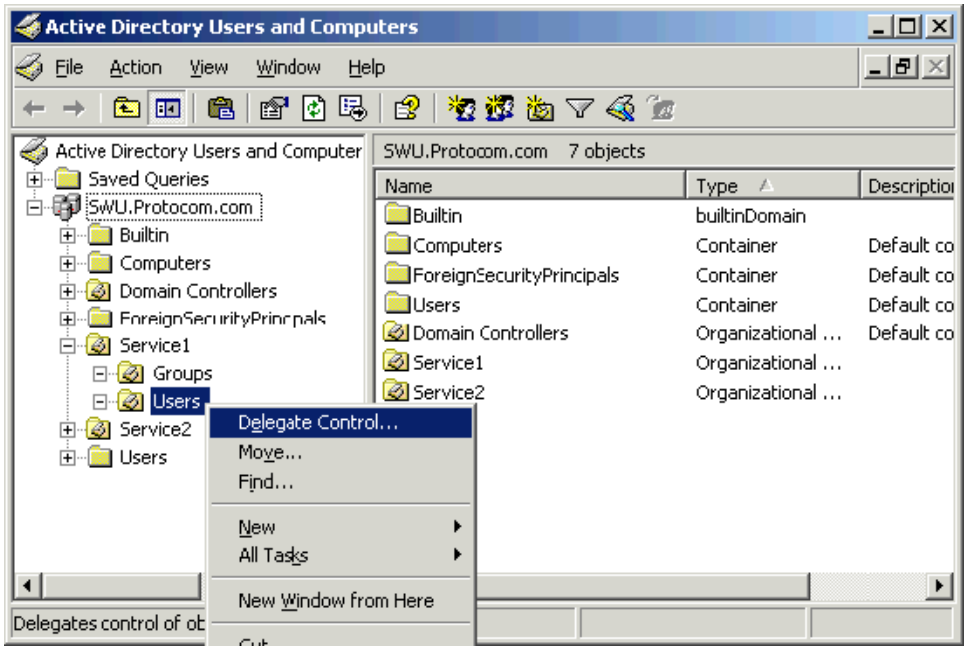
4 Give the group a descriptive name, such as Service1-SecureLogin-Admins.



5 Add the appropriate users to the group.

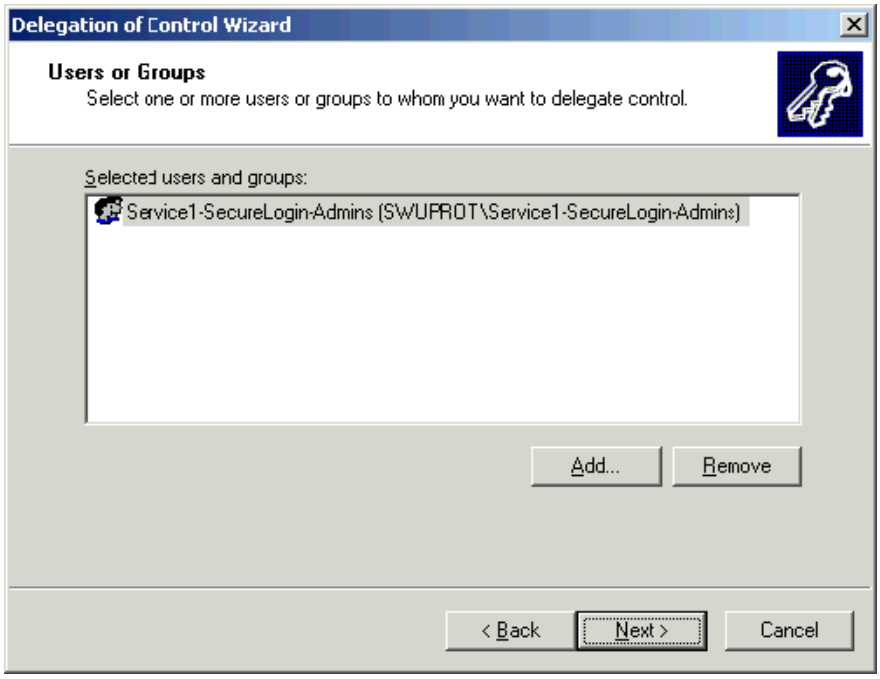


6 Delegate the permissions to the SecureLogin attributes at the container where the users are.

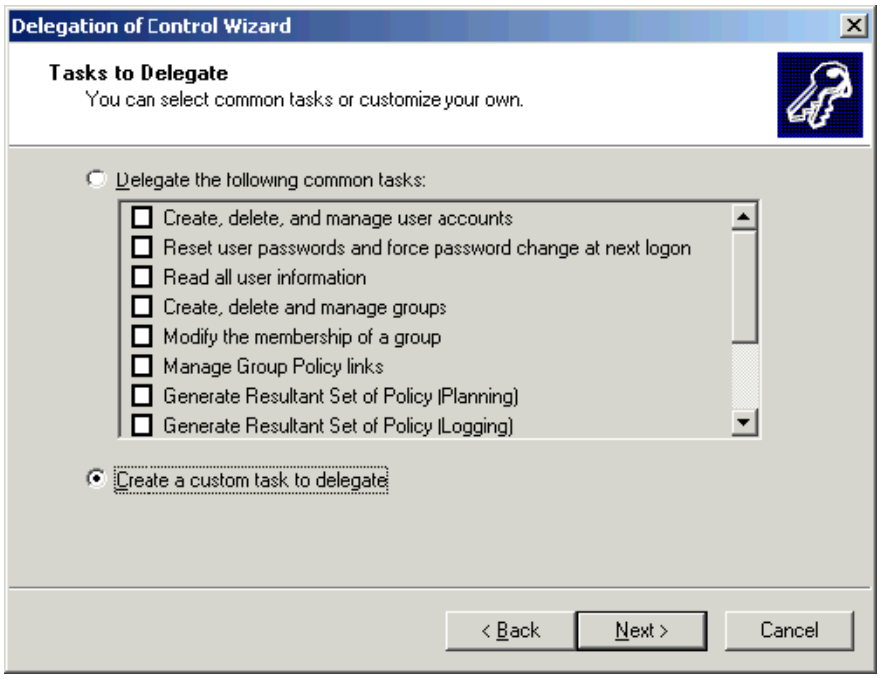


7 The Delegate Control wizard opens.

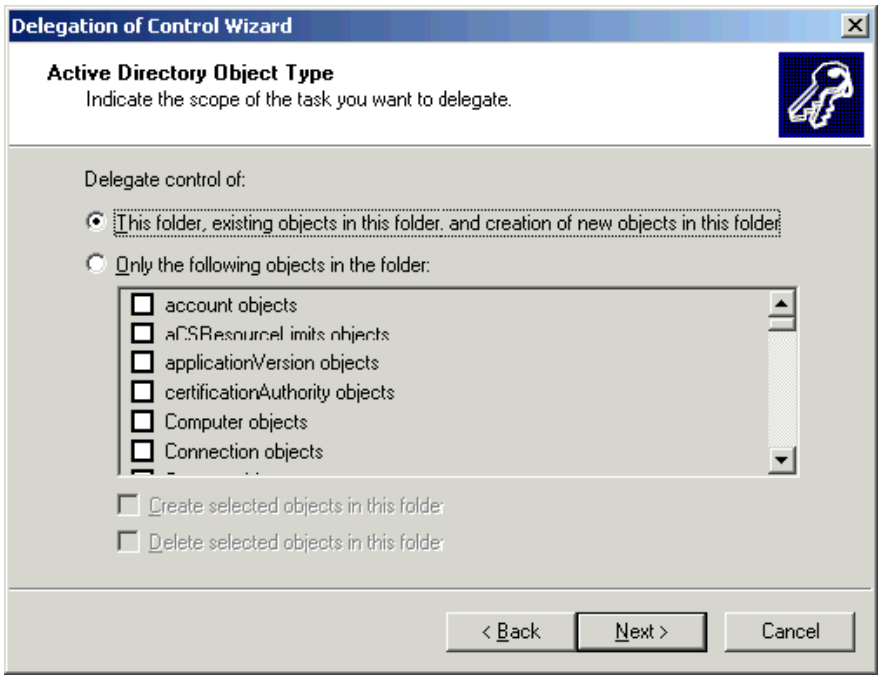
8 Add the group you want to delegate control, then click *Next*.



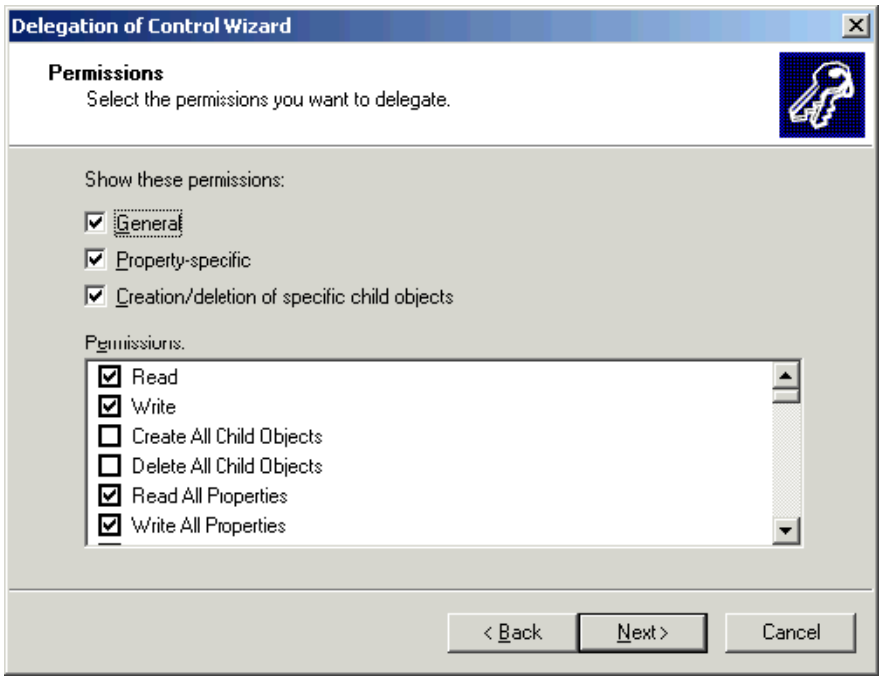
9 Select create a custom task to delegate, then click *Next*.



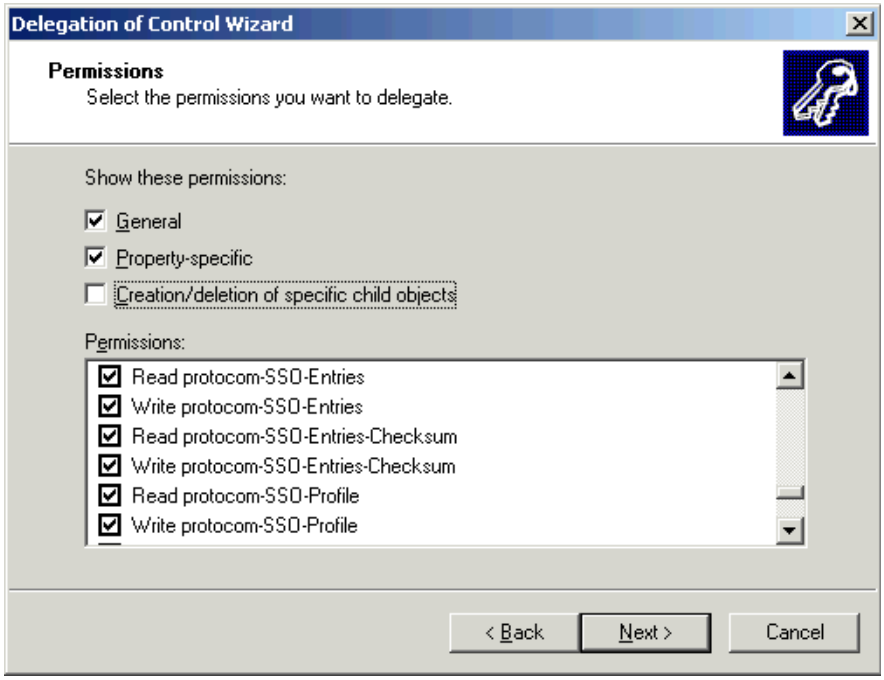
10 Select *This folder, existing objects in this folder, and creation of new objects in this folder,* then click *Next*.



11 Since these are administrator level users they will be granted permissions to manage all aspects of the container and its subordinate objects. Select the *General*, *Property-specific* check boxes. Select the *Read*, *Write*, R*ead All Properties*, and *Write All Properties* permissions.



Verify that you have all Protocom permissions with *Read* and *Write*. Click *Next* to continue.

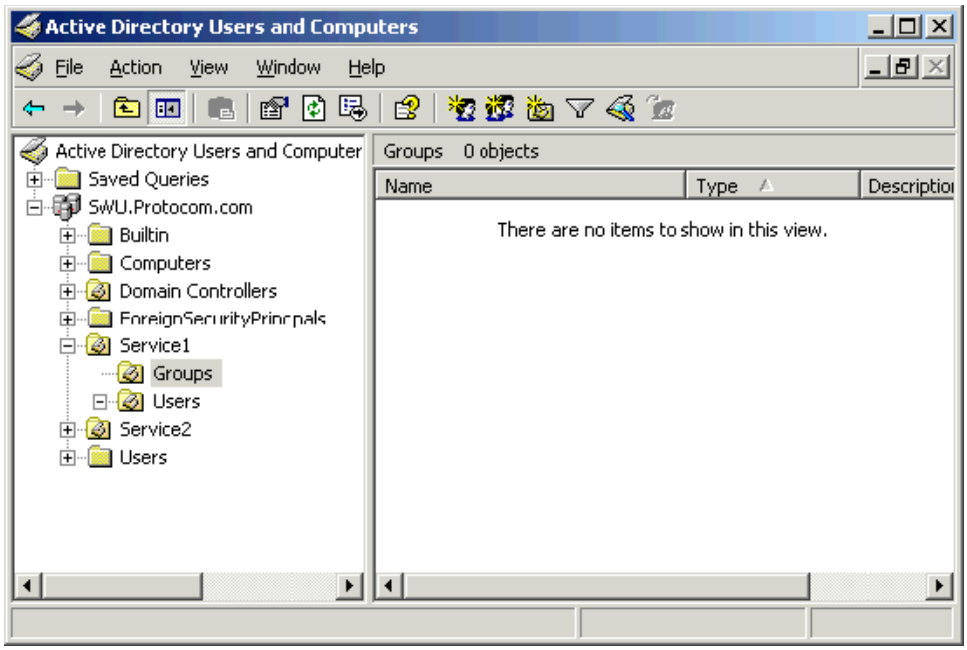


12 You are now finished with the delegate control wizard for the Service1-SecureLogin-Admins group. Click *Finish*.

21.4 Assigning Permissions for SecureLogin Help Desk

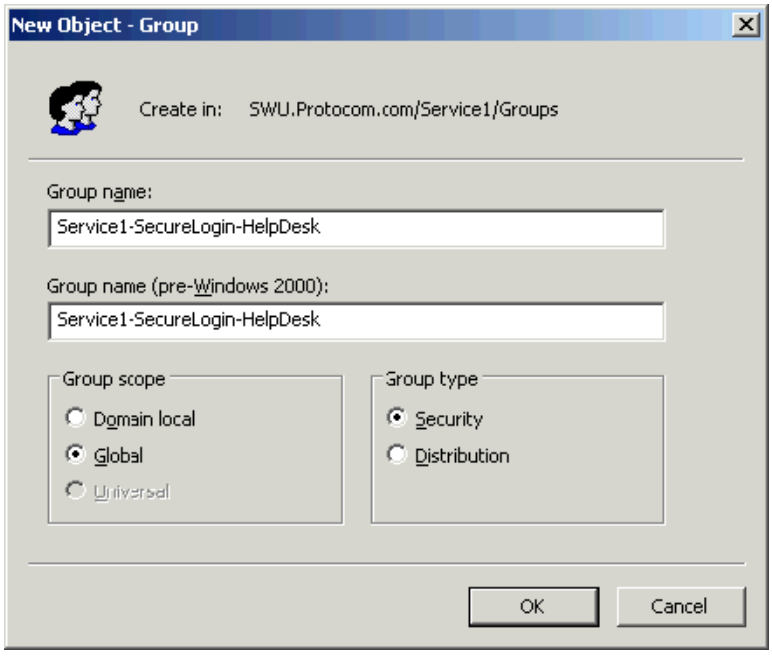
1 Login to the Active Directory domain as a administrative level user.

2 On a workstation or server open Active Directory User and Computers, and browse to the OU where you would like to create the group that will hold the Help Desk users who will work with SecureLogin for the selected container and its children.

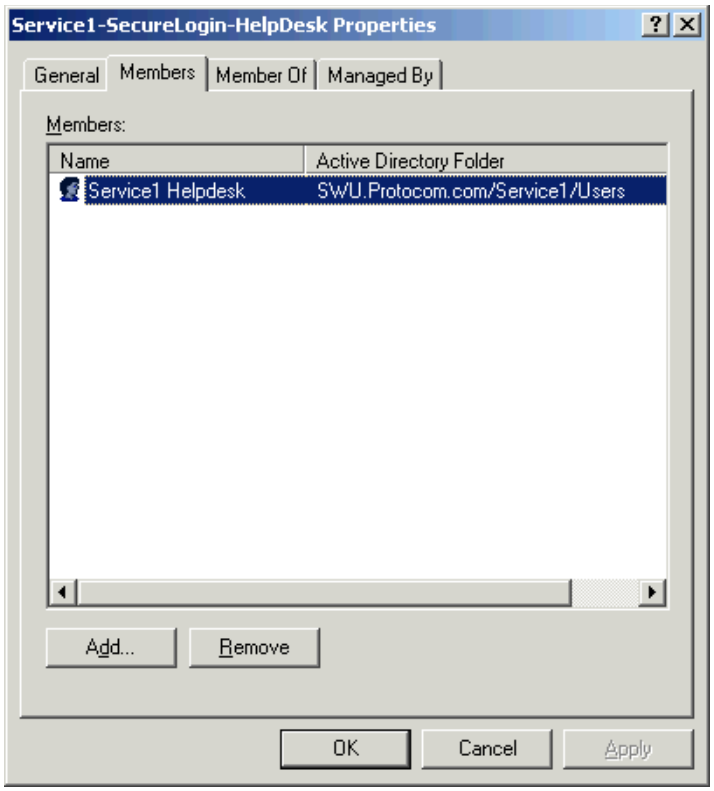


3 Click the create group button 

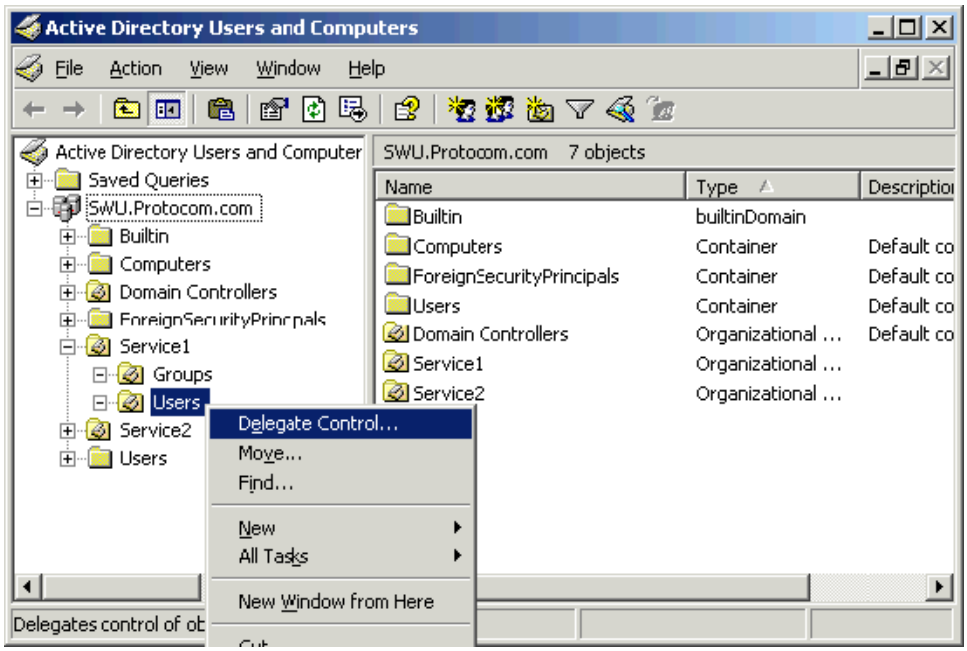
4 Give the group a descriptive name, such as Service1-SecureLogin-Help Desk.



5 Add the appropriate users to the group.

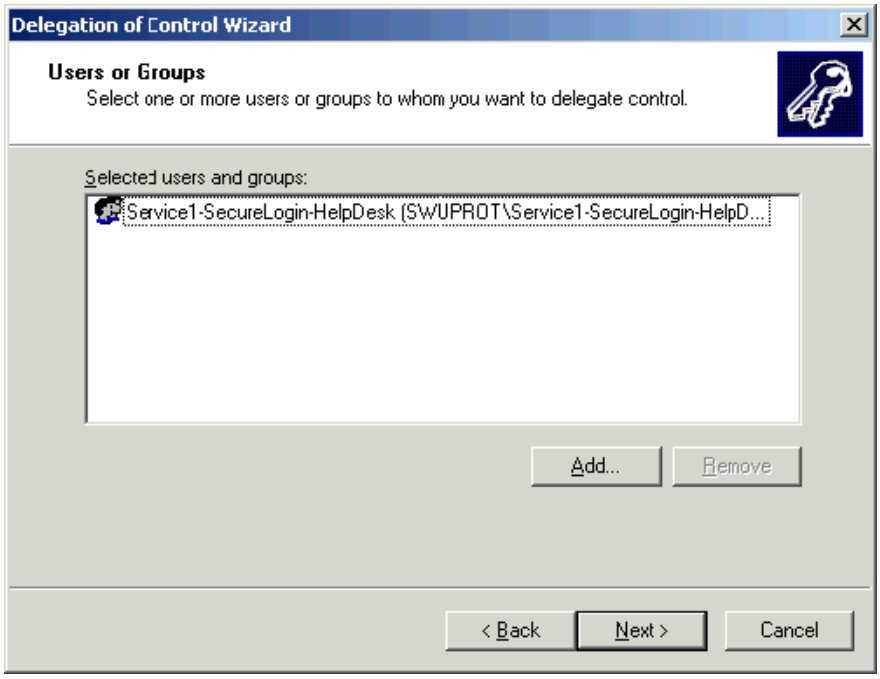


6 Delegate the permissions to the SecureLogin attributes at the container where the users are.

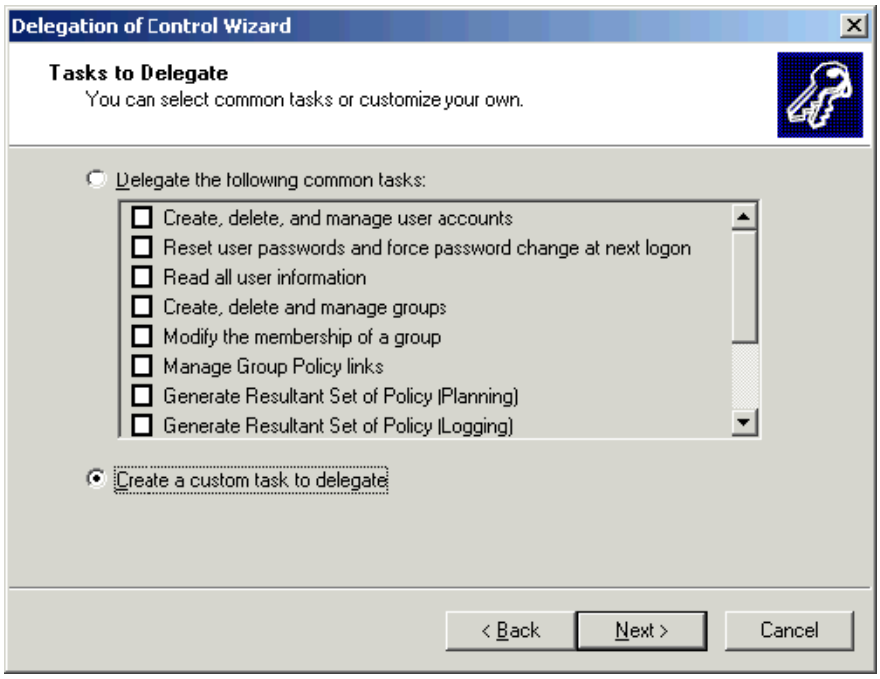


7 The Delegate Control wizard opens.

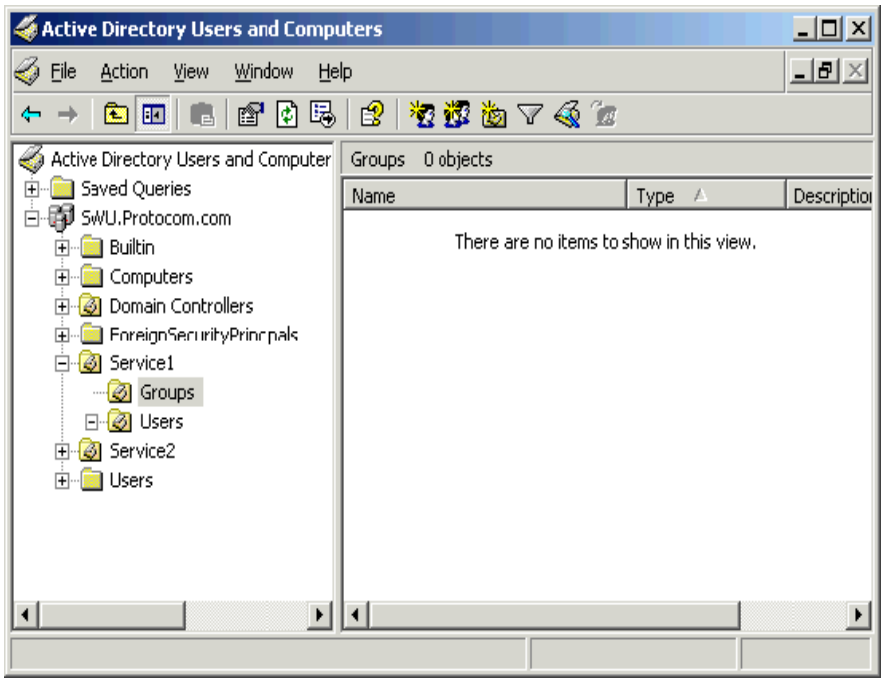
8 Add the group you want to delegate control to, then click *Next*.



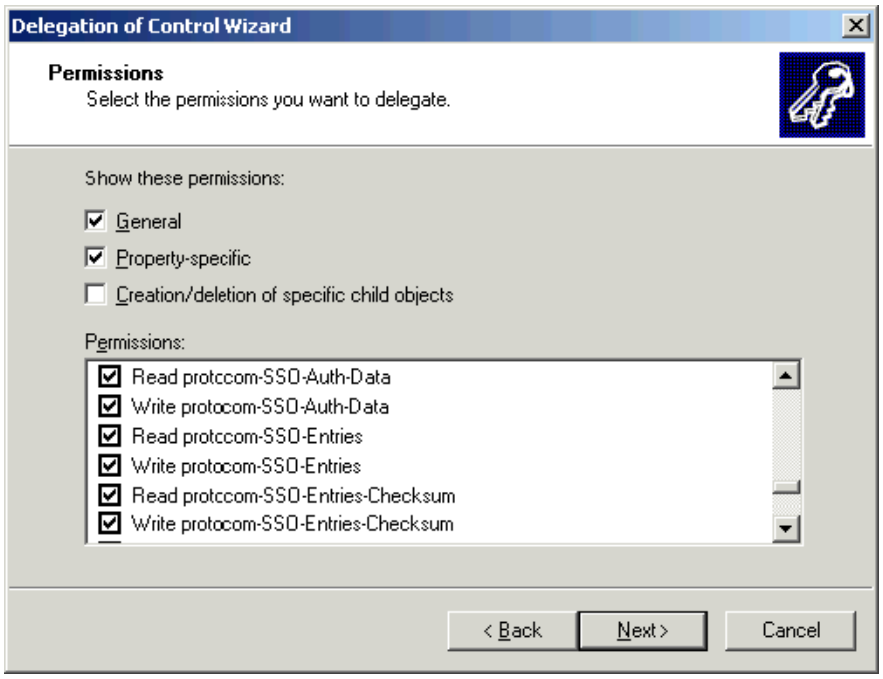
9 Select create a custom task to delegate, then click *Next*.



10 Select the *Only the following objects in the folder*, then scroll down to user objects and select it. Click *Next*.



11 Since these are SecureLogin Help desk level users they will only be granted permissions to manage the SecureLogin attributes. Select the General and Property-Specific checkboxes. Then scroll down and select both the read and write permissions for all protocom- attributes.



12 You are now finished with the delegate control wizard for the Service1-SecureLogin-Admins group. Click *Finish*.

21.5 Assigning SecureLogin Client Settings for Administrators and Help Desk Groups

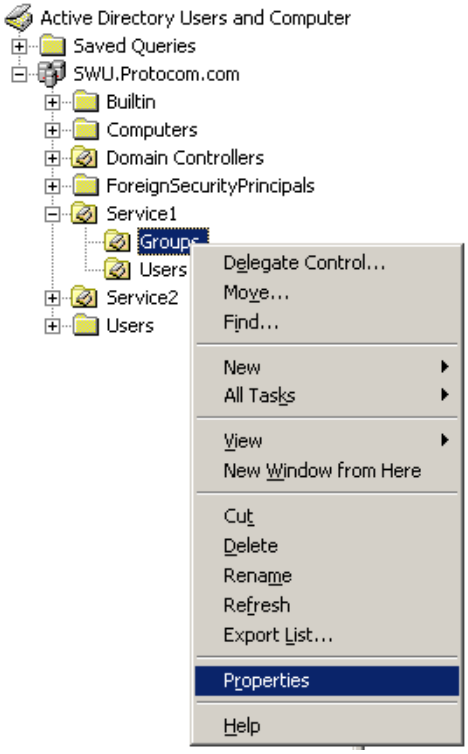
Now that you have assigned the correct Directory permissions to allow members of the administrators and help desk groups to read and write the protocom attributes, you need to assign the SecureLogin client settings (SecureLogin preferences) to allow them to see what they have permissions to access. This is required to override the more restrictive settings the user will inherit from their parent container.

To accomplish this, you can either directly modify the users individual settings. A viable approach if you have a few users who will be granted the elevated permissions. This said, many customers still choose the direct assignment approach, as it can reduce the steps when troubleshooting where someone is getting a specific client setting from. Alternatively, you might utilize SecureLogin's support for group policies. In either case, please see step 8 in this section of the document for the recommended settings.

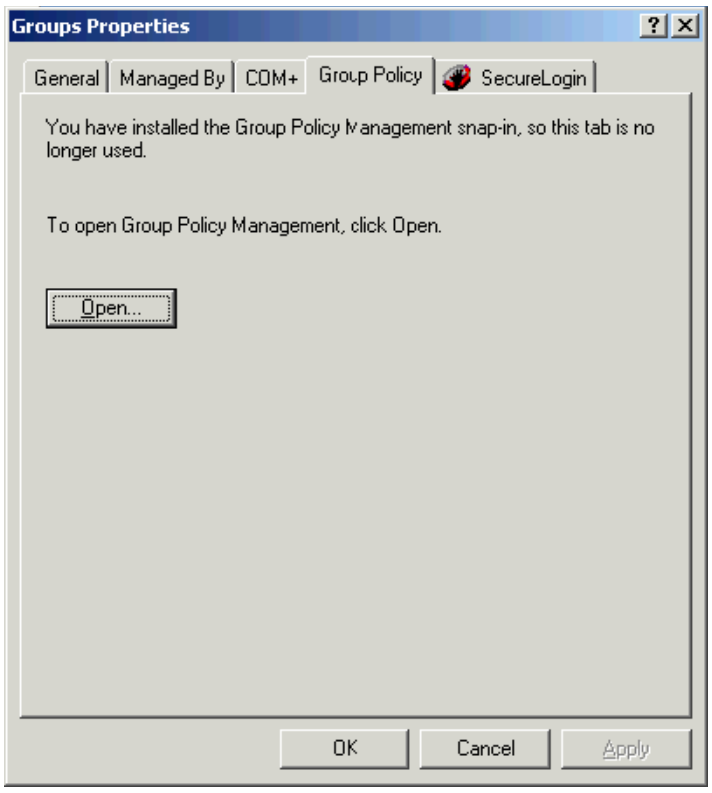
For the sake of this document, it will be assumed you know how to assign individual user's settings, and thus this document will focus on the use of group policies (assuming the feature was enabled during the product installation). As stated previously, both methods have their merits and should be evaluated before deciding on an approach.

21.5.1 Creating the Group Policy

1. Login to the Active Directory domain as a administrative level user.
2. On a workstation or server open Active Directory User and Computers, and browse to the OU that contains the groups that you created earlier. Right click it, select *Properties*.



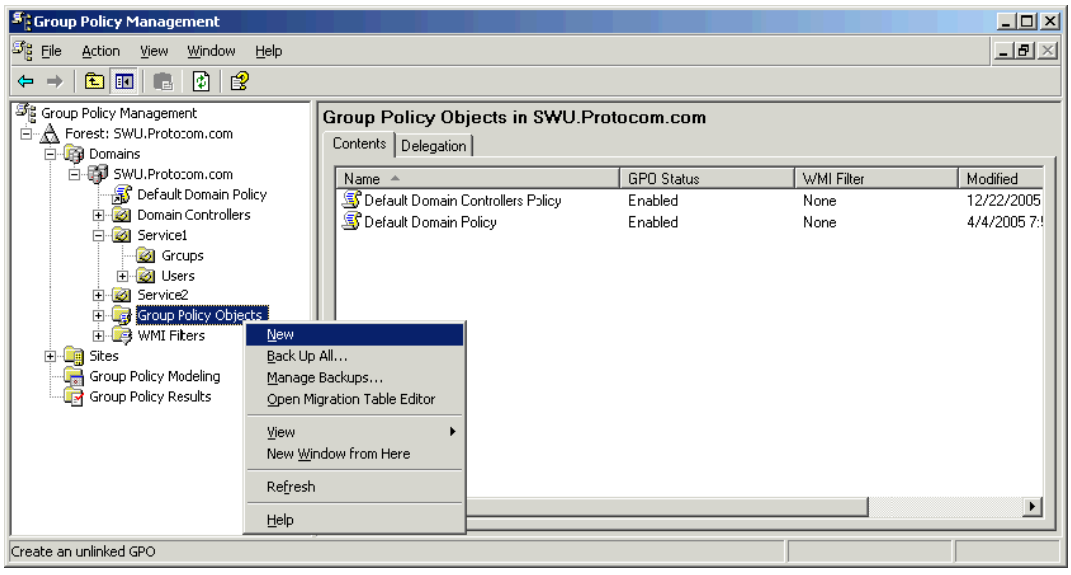
3 In the properties dialog that opens up, select the *Group Policy Tab*.



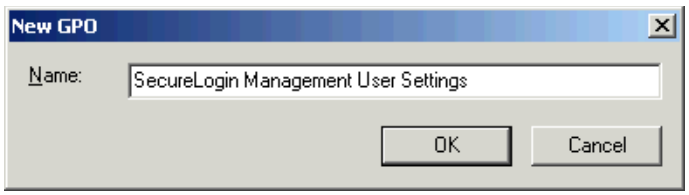
NOTE: In this example the Group Policy Management snap-in has been installed. It can be downloaded from Microsoft (http://www.microsoft.com/downloads/details.aspx?FamilyID=0a6d4c24-8cbd-4b35-9272-dd3cbfc81887&displaylang=en)

---

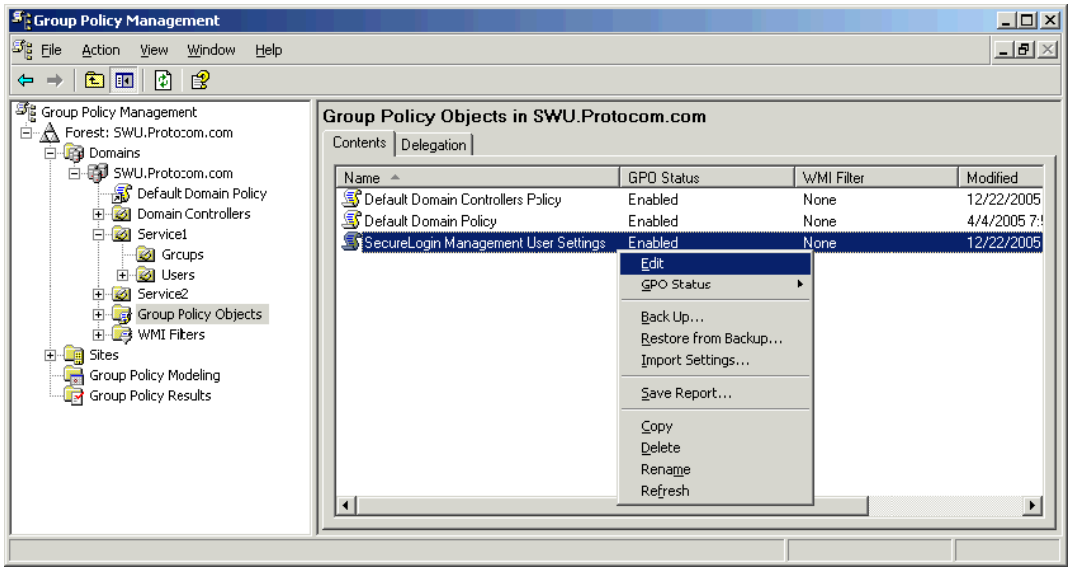
4 Click the *Open* button, the Group Policy Management (GPM) interface will open. Select the *Group Policy Objects* container and right click it. Select *New*.



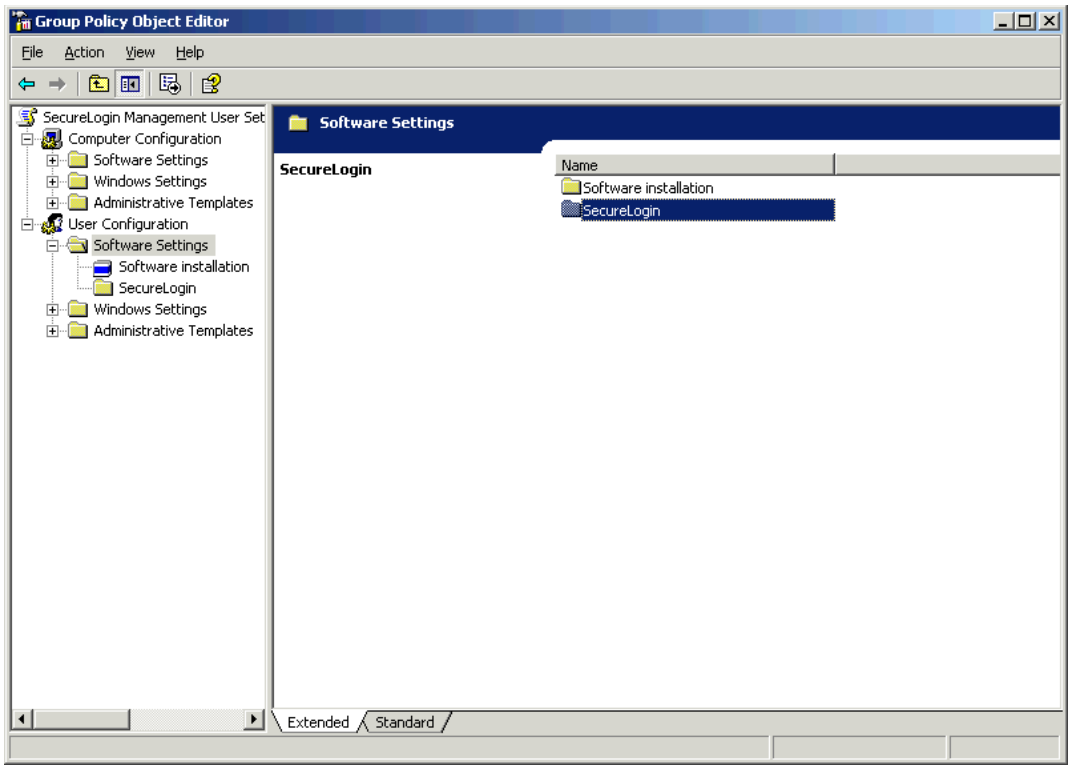
5 Enter a name for the GPO.



6 Right click the new GPO and select *Edit*.



7 Browse to the *User Configuration* > *Software Settings*. In the right hand pane, double click SecureLogin. The SecureLogin management interface will open up.



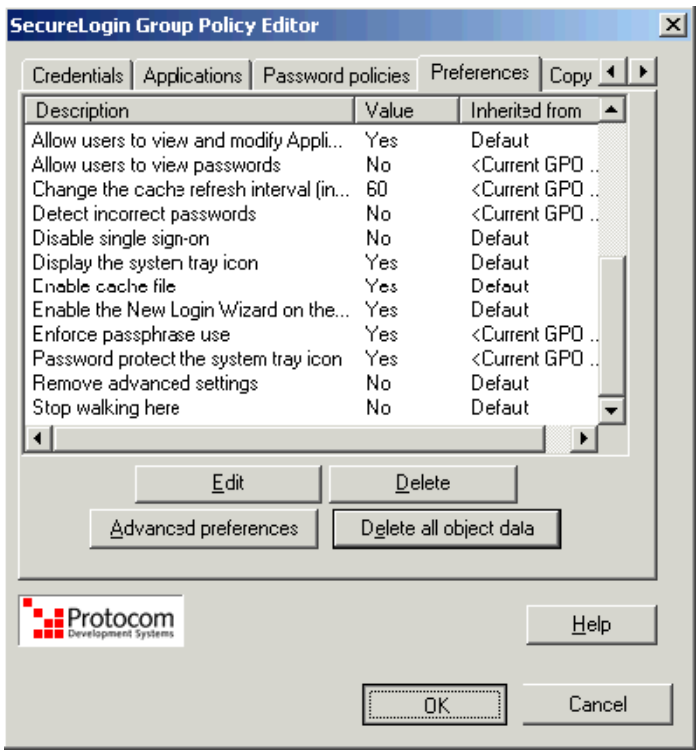
8 In the SecureLogin management interface, select the *Preferences* tab. Set each setting in accordance with what you want the users to do.

NOTE: The users referred in this document are administrators and help desk staff. They have full access to the SecureLogin client. Your configuration might differ slightly.

The preferences highlighted are the one that are critical to ensure users are able to mange SecureLogin. Ensure they are set as shown in the figure.

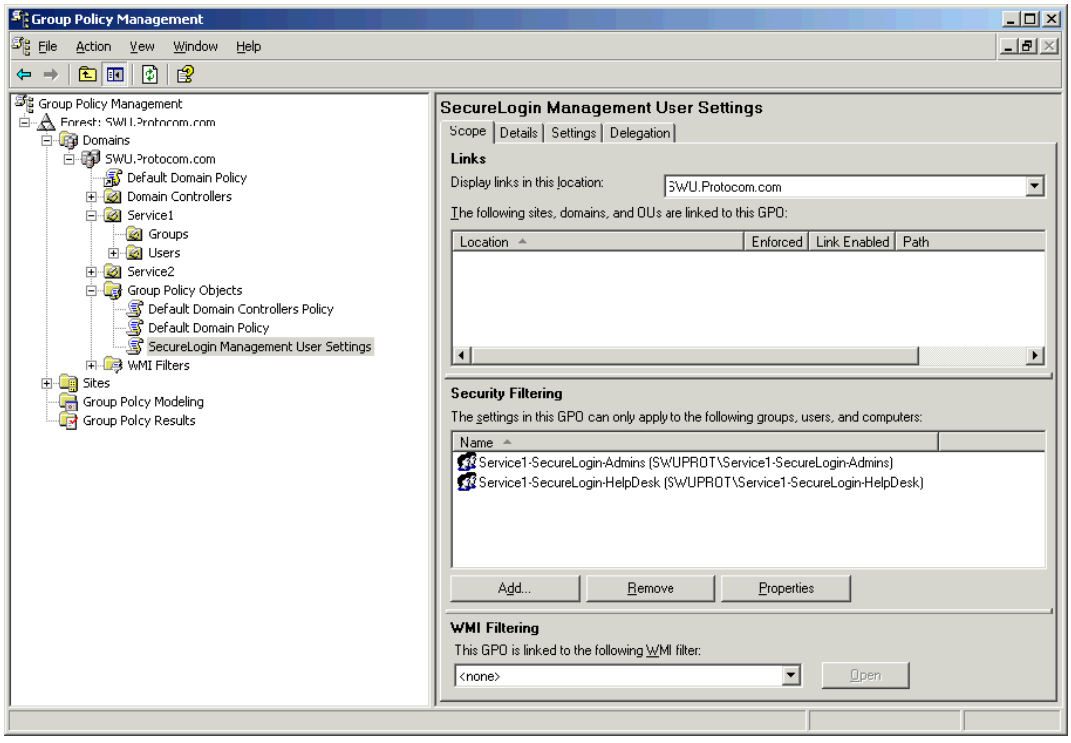
Add application prompts for Internet ...	No	<Current GPO .
Add application prompts for Java app...	No	Default
Add application prompts for Windows...	No	<Current GPO .
Allow single sign-on to Internet Explorer	Yes	Default
Allow single sign-on to Java applicati...	Yes	Default
Allow single sign-on to Netscape	No	<Current GPO .
Allow single sign-on to Windows appl...	Yes	Default
Allow user to backup/restore	Yes	Default
Allow users to modify User ID descrip...	Yes	Default
Allow users to view and change Pref...	Yes	Default
Allow users to view and modify API p...	No	<Current GPO .
Allow users to view and modify Appli...	Yes	Default
Allow users to view passwords	No	<Current GPO .
Change the cache refresh interval (in...	60	<Current GPO .
Detect incorrect passwords	No	<Current GPO .
Disable single sign-on	No	Default
Display the system tray icon	Yes	Default
Enable cache file	Yes	Default
Enable the New Login Wizard on the...	Yes	Default
Enforce passphrase use	Yes	<Current GPO .
Password protect the system tray icon	Yes	<Current GPO .
Remove advanced settings	No	Default
Stop walking here	No	Default

9 Click *Ok* on the SecureLogin management interface. This might take a minute to save.

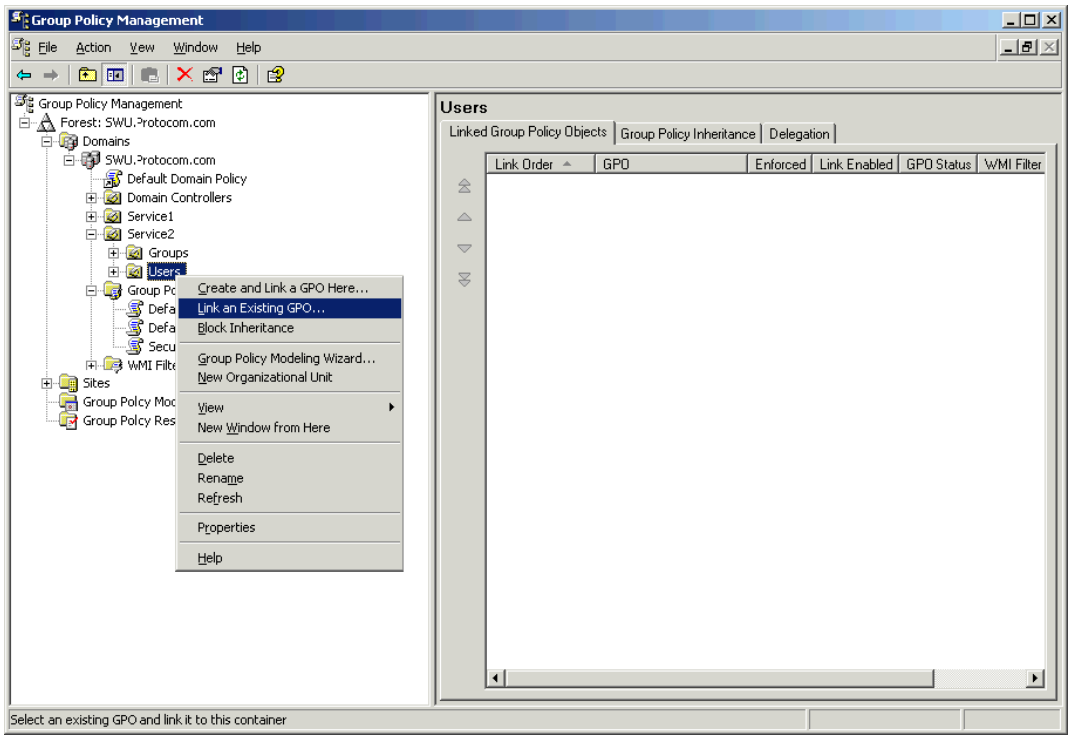


10 Close the GPO editor.

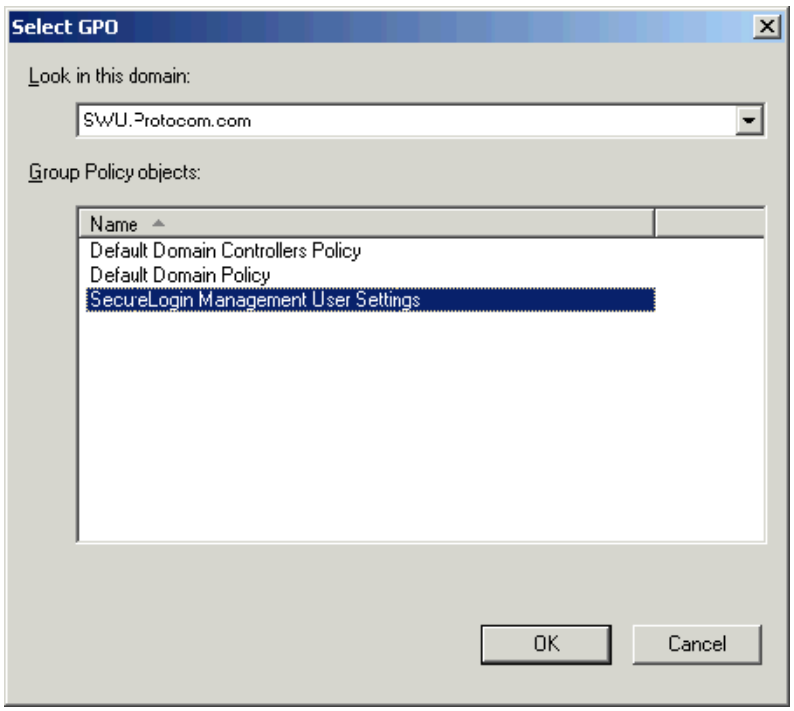
11 In the GPM, select the new GPO you created, remove the Authenticated Users group, and add the admin and help desk groups you created in the previous two sections.



12 Link this policy to the OU where the users are located. Right click and select *Link to an existing GPO*.



13 Select the GPO you created, click OK.



14 Close the GPM. Click *OK* on the group policy tab.

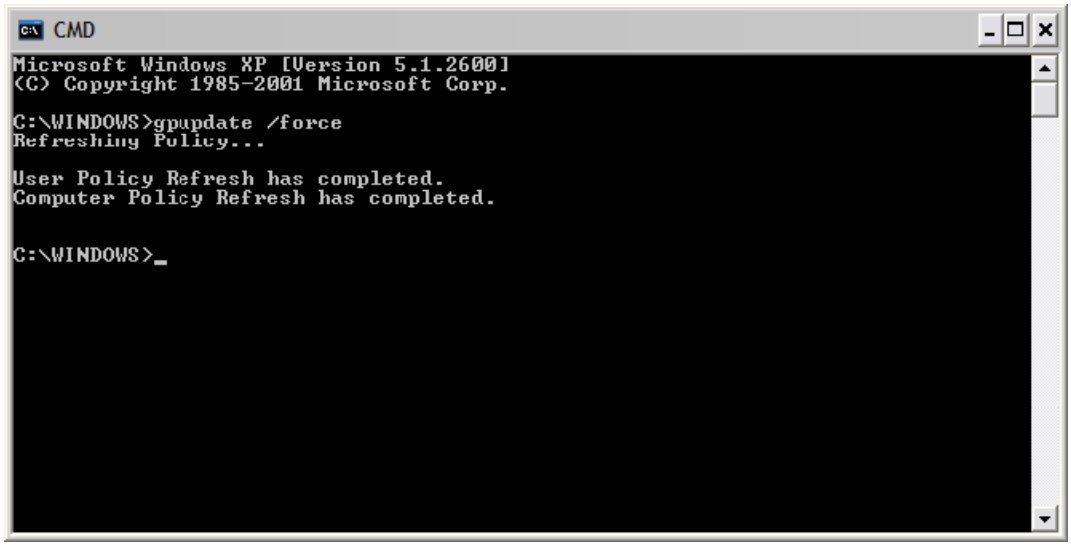
15 Close Active Directory Users and Computers.

21.5.2 Testing your configuration

If you chose to use individual assignment or GPO assignment, proceed with the following tests to confirm your updated configuration

1 On a workstation with SecureLogin and the Active Directory Admin Pak, login as a user who is a member of one of the groups you have configured as SecureLogin administrators or help desk.

2 If your GPO refresh has not occurred, you can manually force the update by going to a command line and issuing the gpupdate /force command (Windows XP). You should see results similar to the following:



3 Launch Active Directory Users and Computers. Navigate to the container where you delegated control. As a member of the Admins group you should be able to manage the OU's, and subordinate objects, applications and preferences.

As a member of the Help Desk group you should be able to only make changes to the users in the OU. It might appear that as a help desk user you can save changes to the OU, but that is not the case. And if you close the Single Sign-On properties and then open it back up, you will see the changes were not saved.

Administering Desktop Automation Services

22

The ARS.exe is the center of Desktop Automation Services. You can configure this object with an independent set of instructions by using an XML document that is obtained through an entry in the Windows registry. The XML document can be obtained either locally on the workstation or through the directory services. The XML document is called the action file and the file is named actions.xml.

22.1 Overview

Each action is a set of configurable user-level operations such as mapping a drive, testing for establishing an authenticated connection to a directory, and running or shutting down an application. The flexibility of the code to test for conditions or have the action triggers such as hot keys provides tremendous flexibility to change the behavior of the workstation to fit your needs.

After the first action is invoked, the ARSControl.exe service starts up and runs as a Windows service. The ARSControl.exe then parses the actions.xml file and stores the configuration in memory. All actions performed by the ARS.exe and ARSControl.exe are recorded in a DASlog.txt log file at different configurable levels of details.

After you have configured the ARS.exe object, its actions are available individually or in combination from any scripting interface that is available on Windows, for example, VBScript*, JavaScript*, login scripts, and batch files.

22.2 Actions and Description

Each instance of Desktop Automation Services is driven by an XML document that describes the available actions.

The following table describes the elements that might be used to compose a Desktop Automation Services XML input document.

Unless otherwise specified, all XML attributes listed for a given element are required for that element.

Table 22-1 *Desktop Automation Services XML Description*

Registry Setting	Description

`application-runner-script`	This is the parent element for an Desktop Automation Services input document.
	
	`application-runner-script` has no attributes.
	
	`application-runner-script` can contain any number of action elements.
	
	`application-runner-script` Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action">`
	
	`<map-drive drive-letter="o:" remote-name="\\192.168.1.255\sys"/>`
	
	`</action>`
	
	`</application-runner-script>`

`action-triggers`	This element is a parent (container) for action-trigger elements (`on-nds-login`, `on-hot-key`.
	
	`action-triggers` enables Desktop Automation Services executables to respond to workstation events by triggering specified actions as defined in the input document.
	
	`action-triggers` has no attributes.
	
	`action-triggers` can contain any of the following child elements:
	
	◆ on-inactivity-timer
	◆ on-nds-login
	◆ on-ldap-login
	◆ on-hot-key
	◆ on-screen-saver
	◆ on-cardmon
	
	`action-triggers` Example:
	`<action-triggers>`
	
	`<on-nds-login action-name="LoginInAction" tree="NCCD_TREE_1"/>`
	
	`</action-triggers>`

Registry Setting	Description

on-inactivity-timer	This command element provides information to the Desktop Automation Services on the action to be performed if the workstation is inactive for more than the specified period of time.
	
	At the end of the countdown period, a specified action such as `Close all programs` or `Lock the Workstation` is invoked. If a mouse or keyboard action is detected, the countdown timer stops and resets until the next inactivity is detected.
	
	NOTE: The on-inactivity-timer functions only if the network login is present.
	
	`run-application` Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action">`
	
	`<run-application application="notepad.exe" interval="500" serial="true" parameters="" />`
	
	`</action> <action-triggers>`
	
	`<on-inactivity-timer interval1="20"`
	
	`message box="Your workstation will be locked in 5 seconds"`
	
	`interval2="5"`
	
	`action-name="WS-lock"/> </action-triggers>`
	
	`</application-runner-script>`
	
	◆ interval 1 is the time of executing an action.
	◆ interval 2 is the interval after the display of the first warning dialog.
	◆ messagebox contains the message to be displayed in the warning.
	◆ action-name is the name of the action to be executed.
	
	NOTE: You must specify only numbers for interval values in the syntax. If you specify special characters in the action.xml file, it does not behave as expected.
	
	The `on-inactivity-timer` is implemented to work with positive numbers, within a range. If negative or special characters are specified, it behaves erroneously.
	
	Specify the inactivity timer in seconds. For example, 10 seconds.

Registry Setting	Description

on-nds-login	This element defines an action trigger to poll for a workstation user logging in to the eDirectory™ instance identified by the tree attribute. The authentication is through the Novell® Client32™ GINA. If a user logs in to the tree, an action trigger invokes Desktop Automation Services. It tests the primary connection to see if the current tree matches the configuration. If it matches, then Desktop Automation Services executes the configured action identified by the action name attribute value.
	
	on-nds-login element must be contained by an action-triggers the parent element.
	
	on-nds-login has two attributes:
	
	◆ **action-name:** The name of an action defined in the input document that is executed when a user logs in to the tree named by the tree attribute. The action-name must be contained in double quotes.
	◆ **tree:** Connections are tested periodically to see if they are linked to the tree named by this network name. The tree name must be contained in double quotes.
	
	on-nds-login Example:
	`<action-triggers>`
	
	`<on-nds-login action-name="LoginInAction" tree="NCCD_TREE_1"/>`
	
	`</action-triggers>`

on-ldap-login	This element defines an action trigger to poll for a workstation user logging in to the directory through the Novell SecureLogin identified by the server attribute.
	
	Desktop Automation Services tests the primary connection to check whether the current server matches the server attribute specified in the configuration. If the current server matches the configuration, then Desktop Automation Services executes the configured action identified by the action-name attribute value.
	
	on-ldap-login must be contained by an action-trigger parent element.
	
	on-ldap-login has two attributes:
	
	◆ **server:** The connections are tested periodically to check if they are linked to the tree named by this name.
	◆ **action-name:** The name of an action defined in the input document that is executed when a user logs in to the tree named by the tree attribute.
	
	on-ldap-login Example:
	`<action-triggers>`
	
	`<on-ldap-login action-name="LoginInAction" server="192.168.1.255"/>`
	
	`</action-triggers>`

Registry Setting	Description
on-hot-key	This element installs an action trigger. The action trigger responds to the user typing the specified hot key sequence (see the example below) by invoking Desktop Automation Services to execute the input document action that has the same name as the action-name attribute value.on-hot-key elements must be contained by an action-triggers parent element. on-hot-key has three attributes: ◆ **virtual-key:** The hex value of the key based on the virtual key map. This element specifies that it is the second component of the hot key sequence. ◆ **modifiers:** The modifiers indicate the keys that are pressed in together with the virtual-key to cause the hot-key event. The hex value might be a combination of one or more of the following, separated by a plus sign (+): ◆ alt indicates the Alt key ◆ ctrl indicates the Ctrl key ◆ shift indicates the Shift key ◆ win indicates the Windows key This element specifies that it is the first component of the hot key sequence. ◆ **action-name:** The name of an action defined in the input document that is executed when the hot-key sequence is detected. **on-hot-key Example:** <action-triggers> <on-hot-key virtual-key="h" modifiers="ctrl+shift" action-name="HKeyAction"/> </action-triggers> A virtual-key value of 'h' and a modifiers value of 'ctrl+shift' produces a Control-Shift-H HotKey sequence.

Registry Setting	Description
on-screen-saver	This element causes an action to be called when the workstation enters the screensaver mode. on-screen-saver elements must be contained by an action-triggers parent element. on-screen-saver has the following attributes: ◆ **action-name:** The name of the action defined in the input document that is executed when the workstation has entered the screensaver mode and that the specified interval has elapsed. ◆ **interval:** This is the amount of time in milliseconds that the ARSControl waits before running the specified action after a screensaver event is triggered. **NOTE:** To activate this trigger, you must have a Windows system screen saver selected. Set the screen saver wait time to the desired time interval before the workstation activates the screen saver. If you are using DAS to activate the screen saver through the on-inactivity-timer action trigger, set the wait time to a longer timer interval than what you set for the on-inactivity-timer action trigger. For example, you can set the on-inactivity-timer interval to 60 minutes. The screen saver will be triggered from DAS on the shared workstation. **on-screen-saver Example:** <action-triggers> <on-screen-saver action-name="logoff" interval="60000"/> </action-triggers> This results in the logoff action executed in 60 seconds after the Windows screen saver is activated.
on-cardmon	The on-cardmon specifies the action to be performed when a smart card is removed. It specifies an ARSaction to be taken when a card is removed. If a user is logged in through a smart card and logs out due to some security reasons, a specific action like system lock must be performed to ensure that the workstation security is not at risk. on-cardmon element must be contained by an action-trigger parent element. on-cardmon has the following attribute: ◆ **action-name:** The name of the action defined in the input document that is executed when the card is removed from the workstation. **on-cardmon Example:** <action-triggers> <on-cardmon action-name="Log out of the workstation"/> </action-triggers>

Registry Setting	Description

action	This is the parent element for all the commands that constitute an action.
	
	action has one attribute:
	
	name: The name can be any arbitrary string value. The character case in the name used by a caller to invoke an action must match the case used where the action is defined. The action-name must be contained in double quotes.
	
	multi-delay: This command element specifies the interval in executing the same action, twice.
	
	multi-delay Example:
	
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action">`
	
	`<action name="ctrl+l" multi-delay="4000">`
	
	`</action>`
	
	`</application-runner-script>`
	
	action can contain any number of the following child elements:
	
	◆ Hide-Desktop and Unhide-Desktop
	◆ run-application
	◆ pause
	◆ test-app-running
	◆ kill-app
	◆ kill-all-apps
	◆ map-drive
	◆ map-home-drive
	◆ map-location-drive
	◆ test-logged-in
	◆ test-ldap-logged-in
	◆ nds-logout
	◆ ldap-logout
	◆ screen-saver-on
	◆ test-nds-attr-val
	◆ test-ip-subnet
	◆ test-env-variable
	◆ message-box
	◆ execute-user-action
	
	action Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	`<application-runner-script>`
	
	`<action name="sample-action">`

Registry Setting	Description
`Hide-Desktop` and `Unhide-Desktop`	The `Hide-Desktop` and `Unhide-Desktop` action hides and shows the desktop icons and other programs before a user logs in. After the user has logged in, the <on-login> condition is met and the Unhide-Desktop action is invoked to display the hidden icons and programs. **NOTE:** These actions are primarily for a kiosk approach without role-based access or for workstation policies managed through ZENworks® syntax. If you specify special characters in the action.xml file, it does not behave as expected.

`run-application`	This command element provides information that enables Desktop Automation Services to run an application and respond when the application is closed.
	
	◆ **application:** The name of the application to launch, for example, notepad.exe. For unregistered programs, the complete application path and extension must be provided. This launches the application correctly.
	◆ **parameters:** A parameter lists the strings to pass to the application. For example, a normal DOS parameter syntax.
	◆ **serial:** If an application is launched using this command with the serial set to true (in synchronous mode), then the execution of the parent action does not continue until the application is closed or the interval timeout has expired.
	◆ **interval:** The timeout interval is used only when if the serial is true. If the application has not returned by timeout, then Desktop Automation Services stops waiting for a return and executes the action.
	
	`run-application` has one optional attribute:
	
	◆ **on-exit-action:** When the application started by this element is closed, the specified action is called.
	
	`run-application` cannot have any child elements.
	
	`run-application` Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action">`
	
	`<run-application application="C:\Program Files\Mozilla`
	
	`Firefox\firefox.exe" parameters="" on-exit-action=”launchSomethingElseAction”`
	
	`serial="true" interval="500"/>`
	
	`</action>`
	
	`</application-runner-script>`

Registry Setting	Description
pause	pause command waits for a specified number of milliseconds before proceeding to perform the next action. **pause Example:** <?xml version="1.0"?> <!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"> <application-runner-script> <action name="sample-action"> <pause interval="500'/> <kill-app application="xmlspy.exe"/> </action> </application-runner-script> **NOTE:** The interval is mandatory. When 'sample action' is triggered, it waits for 500 milliseconds before executing kill-app action.

Registry Setting	Description

test-app-running	test-app-running command element provides information that enables Desktop Automation Services to test whether an application is running or not.
	
	test-app-running can have only one attribute:
	
	◆ **application:** The name of the application as it is found in the process list.
	
	Because the test-app-running is a test command, it can contain either one or both of the following child elements:
	
	◆ **if-true:** An element containing the command operations to perform if the test returns a true value.
	◆ **if-false:** An element containing the command operations to perform if the test returns a false value.
	
	test-app-running Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action">`
	
	`<test-app-running application="notepad.exe">`
	
	`<if-true>`
	
	`<kill-app application="xmlspy.exe"/>`
	
	`<kill-all-apps exclude-apps="notepad.exe:xmlspy.exe"/>`
	
	`<map-drive drive-letter="F:" remote-name="\\172.16.5.250\sys"/>`
	
	`</if-true>`
	
	`<if-false>`
	
	`<map-drive drive-letter="G:" remote-name="\\192.168.1.255\sys"/>`
	
	`</if-false>`
	
	`</test-app-running>`
	
	`</action>`
	
	`</application-runner-script>`

Registry Setting	Description
kill-app	kill-app command element provides information that enables Desktop Automation Services to close an application. kill-app has one essential attribute: ◆ **application:** The name of the application to close, as found in the process list. kill-app has one optional attribute: ◆ **interval:** This is the amount of time in milliseconds that Desktop Automation Services waits after sending a close command to the application before killing the process. The default interval value is 1000. kill-app cannot contain any child element. **kill-app Example:** <?xml version="1.0"?> <!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"> <application-runner-script> <action name="sample-action"> <kill-app application="xmlspy.exe"/> </action> </application-runner-script>

Registry Setting	Description
kill-all-apps	This command element provides information that enables Desktop Automation Services to kill all the running applications except those specified in the exclude-apps. kill-all-apps has one essential attribute: ◆ **exclude-apps:** The names of the applications that must not be killed. The application names are separated by a colon (:) character. The name of an application listed in this attribute must match the name of the application listed in the *Processes* tab of the Task Manager. kill-all-apps has one optional attribute: ◆ **interval:** The amount of time in milliseconds that Desktop Automation Services waits after sending a close command to an application before killing the process. Because each process is closed in a sequential order, a large interval significantly increases the amount of time the command takes to execute. The default value is 0. kill-all-apps cannot have any child elements. **kill-all-app Example:** <?xml version="1.0"?> <!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"> <application-runner-script> <action name="sample-action"> <kill-all-apps exclude-apps="notepad.exe:xmlspy.exe"/> </action> </application-runner-script>

Registry Setting	Description

map-drive	This command element provides information that enables Desktop Automation Services to do a normal drive mapping.
	
	map-drive has two essential attributes:
	
	◆ **drive-letter:** Specifies the drive letter to assign to the new mapped drive.
	
	◆ **remote-name:** Specifies the UNC file specification for a remote volume to be mapped.
	
	map-drive cannot not contain child elements.
	
	map-drive Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action">`
	
	`<map-drive drive-letter="G:" remote-name="\\192.168.1.255\sys"/>`
	
	`</action>`
	
	`</application-runner-script>`

map-home-drive	This command element provides information that enables Desktop Automation Services to map a drive to a home directory as defined by the homedrive attribute in the user's eDirectory object.
	
	map-home-drive has two essential attributes:
	
	◆ **drive-letter:** Specifies the drive letter to assign to the new mapped drive.
	
	◆ **tree:** Specifies the tree containing the object with the home directory information.
	
	map-home-drive cannot contain any child elements.
	
	map-home-drive Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action">`
	
	`<map-home-drive drive-letter="I:" tree="TestTree"/>`
	
	`</action>`
	
	`</application-runner-script>`

Registry Setting	Description
map-location-drive	This command element provides information that enables Desktop Automation Services to map a drive based on a properties file. map-location-drive has four attributes: ◆ **drive-letter:** Specifies the drive letter to assign to the new mapped volume. ◆ **tree:** Specifies the tree containing the object with the location information. ◆ **attribute:** Specifies the key to be used to obtain a value from the properties file. ◆ **file-name:** Specifies the file system path to a properties file containing information for the map-location-drive operation. This file contains property information in the form of key or value pairs. The property key is located on the left of the equals to symbol (=) in a property item and the value is on the right side. For example: here=\\137.65.60.39\Share2 there=\\137.65.60.39\Share3 map-location-drive cannot contain any child elements. **map-location-drive Example:** <?xml version="1.0"?> <!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"> <application-runner-script> <action name="sample-action"> <map-location-drive drive-letter="T:" tree="TestTree2" file-name="c:\yourFile.c" attribute="yourAttribute"/> </action> </application-runner-script>

Registry Setting	Description

test-logged-in	This command element provides information that enables Desktop Automation Services to test whether the user is logged in to a particular eDirectory server or not.
	
	test-logged-in can have only one attribute:
	
	◆ **tree:** The name of the tree for which the logged in state has to be tested.
	
	Because the test-logged-in is a test command, it can contain either one or both of the following child elements:
	
	◆ **if-true:** An element containing the command operations to perform if the test returns a true value.
	◆ **if-false:** An element containing the command operations to perform if the test returns a false value.
	
	test-logged-in Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action">`
	
	`<test-logged-in tree="TestTree">`
	
	`<if-true>`
	
	`<run-application application="explorer.exe" parameters="" serial="false" interval="1000"/>`
	
	`<map-home-drive drive-letter="I:" tree="TestTree"/>`
	
	`</if-true>`
	
	`<if-false>`
	
	`<map-location-drive drive-letter="J:" tree="TestTree" file-name="c:\myFile.c" attribute="myAttribute"/>`
	
	`</if-false>`
	
	`</test-logged-in>`
	
	`</action>`
	
	`</application-runner-script>`

Registry Setting	Description

test-ldap-logged-in	This command element provides information that enables Desktop Automation Services to test whether the user is logged in to a particular LDAP server or not. This command must only be used when using the LDAP GINA and Novell client32 is not used for authentication.
	
	test-ldap-logged-in can have only one attribute:
	
	◆ **server:** The name of the server for which the logged-in state must be tested.
	
	Because test-ldap-logged-in is a test command, it can contain either one or both of the following child elements:
	
	◆ **if-true:** An element containing the command operations to perform if the test returns a true value.
	◆ **if-false:** An element containing the command operations to perform if the test returns a false value.
	
	test-ldap-logged-in Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action">`
	
	`<test-ldap-logged-in server="192.168.1.255">`
	
	`<if-true>`
	
	`<run-application application="explorer.exe" parameters="" serial="false" interval="1000"/>`
	
	`</if-true>`
	
	`<if-false>`
	
	`<run-application application="iexplore.exe" parameters="" serial="false" interval="1000"/>`
	
	`</if-false>`
	
	`</test-logged-in>`
	
	`</action>`
	
	`</application-runner-script>`

Registry Setting	Description
nds-logout	This test command element provides information that enables Desktop Automation Services to log out of the primary NDS® connection. nds-logout cannot not have any attributes. nds-logout cannot have any child attributes. **nds-logout Example:** <?xml version="1.0"?> <!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"> <application-runner-script> <action name="sample-action"> <nds-logout/> </action> </application-runner-script>
ldap-logout	This test command element provides information that enables Desktop Automation Services to log out of Novell SecureLogin. ldap-logout can have one optional attribute: • **gina:** Can have either true or false values. If the value is true, the login dialog box for Novell SecureLogin is displayed after logging out of Novell SecureLogin. If the value is false, no action is taken. The default value is true. ldap-logout cannot have any child elements. **ldap-logout Example:** <action name="logoff"> <pause interval="100"/> <kill-all-apps exclude-apps="slbroker.exe:slwinsso.exe:slproto.exe:explorer.exe:"/> <ldap-logout gina="true"/> </action>

Registry Setting	Description

screen-saver-on	This action tag invokes the Windows screen saver, which triggers the on-screen-saver action. When this action is triggered, the Windows screen saver is started and the DAS on-screen-saver is invoked with timer.
	
	This action locks the workstation and triggers the screen saver, which covers up any icons and browsers. screen-saver-on elements must be contained by an action-triggers parent element.
	
	Use Case: A user is away from the workstation. A pcProx sonar device triggers an event to start the Windows screen saver program. After the defined time interval of inactivity, the user is logged out. In case an activity occurs, the screen saver closes; the user is not logged out. The user returns to the workstation, which is in an undisturbed state. The screen-saver-on action ensures that the icons and browsers are covered.
	
	screen-saver-on has the following attributes:
	
	◆ **action-name:** The name of the action defined in the input document that is executed when the workstation has entered the screen saver mode and that the specified interval has elapsed.
	◆ **interval:** This is the amount of time in milliseconds that the ARSControl waits before running the specified action after a screen saver event is triggered.
	◆ **lock:** If lock is set to true, the workstation is locked after screen saver is activated. The user must enter the password to unlock the workstation and the screen saver.
	
	If lock is set to false, the workstation lock is not activated. Any mouse movement or keystore deactivates the screen saver.
	
	screen-saver-on Example:
	`<action-triggers>`
	
	`<application-runner-script>`
	
	`<action name="Act1">`
	
	`<screen-saver-on/>`
	
	`</action>`
	
	`<action name="Act2">`
	
	`<screen-saver-on lock="true"/>`
	
	`</action>`
	
	`<action name="Act3">`
	
	`<screen-saver-on lock="false"/>`
	
	`</action>`
	
	`<action-triggers>`
	
	`<on-hot-key virtual-key="l" modifiers="ctrl" action-name="Act1"/>`
	
	`<on-hot-key virtual-key="m" modifiers="ctrl" action-name="Act2"/>`
	
	`<on-hot-key virtual-key="n" modifiers="ctrl" action-name="Act3"/>`
	`</action-triggers>`
	
	`</application-runner-script>`

Registry Setting	Description

test-nds-attr-val	This test command element provides information that enables Desktop Automation Services to test whether or not an NDS account contains a particular directory attribute with a particular value.
	
	test-nds-attr-val has four attributes:
	
	◆ **tree:** The name or IP address of the tree containing the account to be searched for the attribute value.
	
	◆ **attr-name:** The name of the attribute to be tested in the NDS account.
	
	◆ **attr-syntax:** The syntax of the attribute to be tested in the NDS account.
	
	The acceptable attr-syntaxes are:
	
	◆ string
	
	◆ integer
	
	◆ boolean
	
	◆ **attr-val:** The value to be searched in the target attribute in the NDS account. The values for the Boolean syntax attribute must be either true or false.
	
	NOTE: If the attribute syntax is string, then the comparison between the value retrieved from the eDirectory and the value of the attr-val is case sensitive.
	
	Because the test-nds-attr-val is a test command, it can contain either one or both of the following child elements:
	
	◆ **if-true:** An element containing the command operations to perform if the test returns a true value.
	
	◆ **if-false:** An element containing the command operations to perform if the test returns a false value.
	
	test-nds-attr-val Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action1">`
	
	`<test-nds-attr-val tree="TestTree" attr-name="cn" attr-syntax="string" attr-val="larry">`
	
	`<if-true>`
	
	`<kill-app application="george.exe"/>`
	
	`<run-application application="fred.exe" parameters="" serial="true" interval="250"/>`
	
	`</if-true>`
	
	`<if-false>`
	
	`<map-drive drive-letter="S:" remote-name="\\172.16.5.253\sys"/>`
	
	`</if-false>`
	
	`</test-nds-attr-val`
	
	`</action>`

Registry Setting	Description

test-ip-subnet	This test command is useful for enabling an action to determine if the workstation resides on a particular network or not. This can be critical if the action is deciding whether to launch a particular application that is available or effective in a given network.
	
	When invoked, the test-ip-subnet command executes the child commands if the current subnet of the workstation and the command's addr attribute value are the same.
	
	test-ip-subnet has two attributes:
	
	◆ **addr:** An IP subnet to compare with the local IP addresses of the machine.
	◆ **subnet:** The subnet mask (in the form of 255.255.255.0) is applied to the addr attribute and the local IP addresses, which are then compared. If the network portion matches, the test returns a true value.
	
	Because the test-ip-subnet is a test command, it can contain either one or both of the following child elements:
	
	◆ **if-true:** An element containing the command operations to perform if the test returns a true value.
	◆ **if-false:** An element containing the command operations to perform if the test returns a false value.
	
	test-ip-subnet Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action">`
	
	`<test-ip-subnet addr="192.168.1.0" subnet="255.255.255.0">`
	
	`<if-true>`
	
	`<run-application application="write" parameters="" serial="true" interval="500"/>`
	
	`</if-true>`
	
	`<if-false>`
	
	`<run-application application="notepad" parameters="" serial="true" interval="500"/>`
	
	`</if-false>`
	
	`</test-ip-subnet>`
	
	`</action>`
	
	`</application-runner-script>`

Registry Setting	Description

test-env-variable	This test command element enables Desktop Automation Services to test whether an environment variable matches a specific value or not.
	
	test-env-variable has two attributes:
	
	◆ **var-name:** The case-sensitive environment variable name. If the variable does not exist, the test returns a false value.
	◆ **var-value:** The value used for case-insensitive comparison with the actual variable value.
	
	Because the test-env-variable is a test command, it can contain either one or both of the following child elements:
	
	◆ **if-true:** An element containing the command operations to perform if the test returns a true value.
	◆ **if-false:** An element containing the command operations to perform if the test returns a false value.
	
	test-env-variable Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action">`
	
	`<test-env-variable var-name="Testvar" var-value="testvalue">`
	
	`<if-true>`
	
	`<run-application application="write" parameters="" serial="true" interval="500"/>`
	
	`</if-true>`
	
	`<if-false>`
	
	`<run-application application="notepad" parameters="" serial="true" interval="500"/>`
	
	`</if-false>`
	
	`</test-env-variable>`
	
	`</action>`
	
	`</application-runner-script>`

Registry Setting	Description
message-box	This command element provides information that enables Desktop Automation Services to display a message box containing the text from the element's caption attribute. message-box has two attributes: ◆ **caption:** The text to be displayed in the dialog box. ◆ **window-name:** The title for the dialog box window. message-box cannot have any child elements. **message-box Example:** <?xml version="1.0"?> <!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"> <application-runner-script> <action name="sample-action"> <message-box caption="HotKey Control+H was pressed." window-name="HotKey Event"/> </action> </application-runner-script>

Registry Setting	Description
execute-user-action	This command directs Desktop Automation Services to access the currently logged-in user and read a custom attribute (ARSUserConfiguration) on that user. The value of this attribute must have the same layout as the standard XML used to configure Desktop Automation Services. **NOTE:** The XML stored in the user object can contain actions. Triggers are not supported. execute-user-action has one attribute: ◆ **action-name:** The name of the configured action read from the user object. **Example value for the ARSUserConfiguration attribute** <?xml version="1.0"?> <!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"> <application-runner-script> <action name="userAction"> <!--. . Any actions may be inserted here. . --> </action> </application-runner-script> **execute-user-action Example:** <?xml version="1.0"?> <!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"> <application-runner-script> <action name="sample-action"> <execute-user-action action-name="userAction"/> </action> </application-runner-script>

Registry Setting	Description

if-true	This is one of the two allowed types of child elements for a test type of command. The other element is “if-false” on page 235.
	
	if-true contains all the commands that must be performed if the test returns a true value. So, if-true can also be a parent element for all the commands that constitute an action.
	
	if-true does not have any attribute values.
	
	if-true can contain any number of the following child elements:
	
	◆ run-application
	◆ test-app-running
	◆ kill-app
	◆ kill-all-apps
	◆ map-drive
	◆ map-home-drive
	◆ map-location-drive
	◆ test-logged-in
	◆ test-ldap-logged-in
	◆ test-nds-attr-val
	◆ test-ip-subnet
	◆ test-env-variable
	◆ message-box
	◆ nds-logout
	◆ ldap-logout
	◆ execute-user-action
	
	if-true Example:
	`<?xml version="1.0"?>`
	
	`<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">`
	
	`<application-runner-script>`
	
	`<action name="sample-action">`
	
	`<test-env-variable var-name="Testvar" var-value="testvalue">`
	
	`<if-true>`
	
	`<run-application application="write" parameters="" serial="true" interval="500"/>`
	
	`</if-true>`
	
	`<if-false>`
	
	`<run-application application="notepad" parameters="" serial="true" interval="500"/>`
	
	`</if-false>`
	
	`</test-env-variable>`
	
	`</action>`
	`</application-runner-script>`

Registry Setting	Description
if-false	This is one of the two allowed types of child elements for a test type of command. The other element is “if-true” on page 234. if-false contains all the commands that must be performed if the test resolves to false. if-false can also be a parent element for all the commands that constitute an action. if-value does not have attribute value. if-value can contain any number of the following child elements: ◆ run-application ◆ test-app-running ◆ kill-app ◆ kill-all-apps ◆ map-drive ◆ map-home-drive ◆ map-location-drive ◆ test-logged-in ◆ test-ldap-logged-in ◆ test-nds-attr-val ◆ test-ip-subnet ◆ test-env-variable ◆ message-box ◆ nds-logout ◆ ldap-logout ◆ execute-user-action **if-false Example:** <?xml version="1.0"?> <!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd"> <application-runner-script> <action name="sample-action"> <test-env-variable var-name="Testvar" var-value="testvalue"> <if-true> <run-application application="write" parameters="" serial="true" interval="500"/> </if-true> <if-false> <run-application application="notepad" parameters="" serial="true" interval="500"/> </if-false> </test-env-variable> </action> </application-runner-script>

22.3 Example XML File

Each instance of Desktop Automation Services is driven by an XML document describing the actions that are available. Following is an example of a Desktop Automation Services input document.

This XML file contains examples of most of the XML elements that can be used to compose action sequences using Desktop Automation Services.

```
<?xml version="1.0"?>
<!DOCTYPE application-runner-script SYSTEM "ARS_1.0.dtd">
<application-runner-script>
  <action name="worksuite">

    <!-- KILL THE GAMES -->
    <kill-app application="freecell.exe"/>
    <kill-app application="winmine.exe"/>
    <kill-app application="sol.exe"/>

    <!-- LOAD THE WORK APPS -->
    <test-app-running application="notepad.exe">
      <if-true>
      </if-true>
      <if-false>
        <run-application application="notepad.exe" on-exit-action="gamesuite"
parameters="" serial="true" interval="500"/>
      </if-false>
    </test-app-running>
    <test-app-running application="calc.exe">
      <if-true>
      </if-true>
      <if-false>
        <run-application application="calc.exe" on-exit-action="gamesuite"
parameters="" serial="true" interval="500"/>
      </if-false>
    </test-app-running>
    <test-app-running application="mspaint.exe">
      <if-true>
      </if-true>
      <if-false>
        <run-application application="mspaint.exe" on-exit-action="gamesuite"
parameters="" serial="true" interval="500"/>
      </if-false>
    </test-app-running>
  </action>
  <action name="gamesuite">

    <!-- KILL THE WORK APPS -->
    <kill-app application="notepad.exe"/>
    <kill-app application="calc.exe"/>
    <kill-app application="mspaint.exe"/>

    <!-- LOAD THE GAMES -->
    <test-app-running application="freecell.exe">
      <if-true>
      </if-true>
      <if-false>
       <run-application application="freecell.exe" on-exit-action="worksuite"
```

```
parameters="" serial="true" interval="500"/>
      </if-false>
    </test-app-running>
    <test-app-running application="winmine.exe">
      <if-true>
      </if-true>
      <if-false>
        <run-application application="winmine.exe" on-exit-action="worksuite"
parameters="" serial="true" interval="500"/>
      </if-false>
    </test-app-running>
    <test-app-running application="sol.exe">
      <if-true>
      </if-true>
      <if-false>
        <run-application application="sol.exe" on-exit-action="worksuite"
parameters="" serial="true" interval="500"/>
      </if-false>
    </test-app-running>
  </action>
</application-runner-script>
```

Security Considerations

23

Consider the following to help ensure security for Novell SecureLogin:

- It is not recommended to use pcProx alone for authentication. Use pcProx in conjunction with other NMAS authentication methods for more security.
- Use the AES encryption standard for the encryption of SecureLogin data.
- Back up SecureLogin data and directory data by using encryption and password protection.
- Use `AAVerify` to provide additional advanced authentication to single sign-on applications with NMAS methods.
- Provide information to users about using a smart card, including details about how to store application credentials on the card, and how to encrypt the directory data store by using PKI-based credentials.
- Protect the SecureLogin desktop shortcut with a password so that others cannot view SecureLogin data.
- Prevent certain SecureLogin settings and options from being visible or modifiable by others.
- Use a universal password for increased security by providing additional layers of policies.
- Require SecureLDAP when using LDAP to authenticate to SecureLogin.
- Use Novell SecretStore® to provide additional security to SecureLogin data stored on eDirectory.
- Use NMAS to provide advanced authentication, such as pcProx, fingerprint, and token-based authentication.
- Store SecureLogin credentials in a PIN-protected smart card, which provides a secure, portable, and efficient single sign-on solution.
- Keep the local cache files in a user profile directory so that only the corresponding Windows user can access them.
- Enable a passphrase to provide additional security to SecureLogin user data.
- Ensure strict password policies for SecureLogin users and for all single sign-on logins. Randomization of passwords and hiding them from end users is also essential.
- Use auditing features like SNMP alerts, Windows event logs, and Novell Audit logging to capture SecureLogin activity wherever applicable.
- When you are using LDAP with NMAS, the Novell SecureLogin universal password must be enabled.

Error Messages

A

SecureLogin error messages display a number code that generally includes a text description of the error. SecureLogin error numbers currently range between -101 and -914. Following is a list of these error message, their cause, and the appropriate action to take.

Some of the codes displayed in SecureLogin error messages are not native SecureLogin codes. Refer to the relevant application's Help for assistance with the following:

Novell eDirectory™: Numbers between -1 and -813

Microsoft Active Directory; Error codes such as, 0x80070002

For more information about Active Directory error codes, go to the Microsoft Web site (http://msdn.microsoft.com)

The Secure Workstation post-login method failed, error: - 1449990268

Possible Cause: NMAS sequence with SecureWorkstation post-login method is created and not configured using iManager plugin for SecureWorkstation.

Action: Novell SecureLogin displays this error if any NMAS sequence with SecureWorkstation post-login method is used without activation.

Before using any sequence with SecureWorkstation post-login method, configure and activate the sequence using Secureworkstation iManager plugin.

-102: BROKER_NO_SUCH_ENTRY

Possible Cause: You tried to load an application definition or variable that does not exist.

For example, you set up Terminal Launcher to run from a shortcut or to run a particular application definition, but the application definition does not exist.

Action: Check that the name of the application definition is actually defined in SecureLogin. Verify that the name is the same as the name specified in the application definition.

-103: BROKER_INVALID_CLASS_CREATED

Possible Cause: Data has become corrupted, or you are running an earlier version.

SecureLogin is trying to create a new version of the application definition data format that was stored in ANDS.

Action: Upgrade the older SecureLogin client to the new client. Install the latest SecureLogin software.

-104: BROKER_CREATE_CLASS_FAILED

Possible Cause: The SecureLogin client has run out of memory.

Action: Free up some memory. Try again later.

-105: BROKER_REMOVE_ENTRY_FAILED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-106: BROKER_UPDATE_GET_ENTRY_FAILED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-107: BROKER_ENTRY_NOT_FOUND

Possible Cause: An attempt to load an application definition or variable that does not exist.

Action: Check that the name of the application definition is actually defined in SecureLogin. Verify that the name is the same as the name specified in the application definition editor.

-109: BROKER_SCRIPT_BUFFER_ALLOC_FAILED

Possible Cause: The SecureLogin client has run out of memory.

Action: Free up some memory. Try again later.

-110: BROKER_NO_MORE_PLATFORMS

Possible Cause: Data is corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-111: BROKER_NO_MORE_VARIABLES

Possible Cause: Data is corrupted or the software is not working as intended.

Action: Contact Novell Support.

-112: BROKER_NO_SUCH_VARIABLE

Possible Cause: You are trying to use an undefined variable.

Because SecureLogin is not prompting you for the variable, data has become corrupted, or some other situation is preventing the software from working as expected.

Action: Contact Novell Support.

-114: BROKER_PRIMARY_NOT_AVAILABLE

Possible Cause: You are not logged on to the directory. You are using the offline cache. Therefore, you cannot perform some directory functions. For example, you cannot change your passphrase.

Action: Log in to the directory.

-116: BROKER_HEADER_DATA_CORRUPT

Possible Cause: Data is corrupted. You might have a customized build for your site, but have installed a standard version of SecureLogin, or have gone from a standard version to a customized build for your site.

Action: Delete the local cache file and try again. If unsuccessful, contact Novell Support.

-120: BROKER_INVALID_PREF_DATA_TYPE

Possible Cause: Data is corrupted or the software is not working as intended.

Action: Contact Novell Support.

-121: BROKER_PREFERENCE_DATA_CORRUPT

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-122: BROKER_TARGET_ENTRY_LIST_NOT_LOADED

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-123: BROKER_CACHE_PASSWORD_INCORRECT

Possible Cause: You have tried to log on from offline mode, but the password you entered does not match the expected password from the local cache.

Typically, the offline password is the passphrase answer.

Action: Enter the correct passphrase answer or directory password.

-129: BROKER_ENTRY_LIST_NOT_NULL

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Delete the local cache file and try again. If unsuccessful, contact Novell Support.

-130: BROKER_ENTRY_LIST_NULL

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Delete the local cache file and try again. If unsuccessful, contact Novell Support.

-131: BROKER_YSM_LIST_NOT_NULL

Possible Cause: Memory is not handled as expected.

Action: Contact Novell Support.

-132: BROKER_SYM_LIST_NULL

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-138: BROKER_SYMBOL_DATA_CORRUPT

Possible Cause: Data has become corrupted in the local cache file or in the directory.

Action: Delete the local cache file and try again. If unsuccessful, contact Novell Support.

-140: BROKER_SCRIPT_DATA_CORRUPT

Possible Cause: Data has become corrupted in application definitions.

Action: Delete the local cache file and try again.

-141: BROKER_PREF_INVALID

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-142: BROKER_SET_PREF_INVALID

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-145: BROKER_SECURITY_ALERT

Possible Cause: Unable to locate security keys (AuthData), but security data appears to exist. It is possible that someone has attempted to gain access to your security data.

Action: Contact your system administrator.

-166: BROKER_INVALID_DES_KEY

Possible Cause: Hex strings are invalid. The DES_KEY variable requires hexadecimal (0-9, A-F) numbers.

Action: Make sure that the DES_KEY variable contains only hexadecimal numbers.

-167: BROKER_INVALID_DES_OFFSET

Possible Cause: Hex strings are invalid. The DES_OFFSET variable requires hexadecimal (0-9, A-F) numbers.

Action: Make sure the DES_OFFSET variable contains only hexadecimal numbers.

-168: BROKER_DESKEY_NOT_FOUND

Possible Cause: You tried to generate a one-time password for a platform. However, you have not defined the DES_KEY variable.

Action: Create the DES_KEY variable.

-169: BROKER_DESOFFSET_NOT_FOUND

Possible Cause: You tried to generate a one-time password for a platform. However, you have not defined the DES_OFFSET variable.

Action: Create the DES_OFFSET variable.

-171: BROKER_CACHE_FILE_OPEN_FAIL

Possible Cause: SecureLogin tried to read or write to the offline cache. However, SecureLogin is unable to open the cache file.

Action: Assign rights so that the specified user object has rights to the cache directory.

-173: BROKER_NO_MORE_CACHE_FILE_DATA

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-174: BROKER_CACHE_SAVE_FAILED

Possible Cause: SecureLogin is unable to save data to the offline cache.

Action: Assign rights so that the specified user object has rights to the cache directory.

-175: BROKER_CACHE_SECRETS_INCORRECT

Possible Cause: The offline cache password is incorrect for either of the following reasons:

- The key used to decrypt the cache file is not the key that the cache file was encrypted with.
- If you log on as a user to a workstation and create a cache file, and then you go to another workstation, reset your passphrase and log on, then when you return to the original workstation this error message appears.

Action: Delete the cache file.

-176: BROKER_PUBLIC_KEY_READ_FAILED

Possible Cause: SecureLogin is unable to read the public key from Active Directory System.

Action: Troubleshoot Microsoft Active Directory System and Microsoft ADAM.

-177: BROKER_PUBLIC_KEY_HAS_CHANGED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-179: BROKER_RTVALUE_DOES_NOT_EXIST

Possible Cause: You tried to read a runtime variable that is not defined.

Action: Check the application definition. Make sure that the variable is set before it is read or used as a command.

-180: BROKER_DS_VARIABLE_NOT_READ

Possible Cause: You used one of the % variables to read a directory attribute, but SecureLogin cannot read the variable.

Action: Make sure that you have spelled the attribute name correctly. Troubleshoot Microsoft Active Directory System or Microsoft ADAM.

-181: BROKER_WRONG_PASS_PHRASE

Possible Cause: The passphrase or password is incorrect. The reason could be:

- You entered the wrong passphrase.
- You tried to change your passphrase, but entered it incorrectly.
- You password protected the SecureLogin notification area icon and entered the incorrect password.

Action: Enter the passphrase or password correctly.

-190: BROKER_NO_AUTH_DATA_FOUND

Possible Cause: Although the SecureLogin Entry attribute has data, the SecureLogin Auth attribute was blank.

Someone deleted the SecureLogin SSO Auth attribute.

Action: Delete the Prot:SSO Entry attribute.

SecureLogin creates these attributes the next time you run SecureLogin.

-192: BROKER_UNABLE_TO_INSTANTIATE

Possible Cause: A module, for example, WinSSO, is unable to connect to the Combroker.

Action: If you are using Windows 95, make sure that you have the latest DCOM update, or reinstall Internet Explorer.

For other platforms, reinstall SecureLogin.

-195: BROKER_FILE_TRAITS_OP_NOT_IMPLEMENTED

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-196: BROKER_DUMMY_OP_NOT_IMPLEMENTED

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-199: BROKER_ERROR_COMMAND_NOT_HANDLED

Possible Cause: An application definition parser encountered an unrecognizable command.

Action: Make sure that:

- The command is spelled correctly.
- The If/EndIf blocks match.

-200: BROKER_END_OF_SCRIPT

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-201: BROKER_UNEXPECTED_END_OF_SCRIPT

Possible Cause: `If/EndIf` or `Repeat/EndRepeat` blocks do not match. SecureLogin reached the end of the application definition without finding an expected `EndIf` or `EndRepeat` command.

Action: Check the application definition. Make sure that the `If/EndIg` and `Repeat/EndRepeat` blocks match.

-206: BROKER_BREAK_BLOCK

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-207: BROKER_END_SCRIPT_NOW

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-210: BROKER_CORPORATE_MOD_ABORTED

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-211: BROKER_ENTRY_ALREADY_ON_LIST

Possible Cause: You tried to add an application definition or variable, but an application definition or variable with that name already exists.

Action: Do one of the following

- Use a different name for the application definition or variable.
- Rename the existing application definition or variable in the application definition editor.

-213: BROKER_NDS_OP_NOT_IMPLEMENTED

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-214: BROKER_UNABLE_TO_GET_CURRENT_OU

Possible Cause: Data has become corrupted or the software is not working as intended.

Action: Contact Novell Support.

-217: BROKER_ARG_NUM

Possible Cause: In application definition language, each command expects a certain number of arguments. You have used either too few or too many arguments for a given command.

Action: Make sure you are passing the correct number of arguments to the command.

-219: BROKER_NOT_A_NUMBER

Possible Cause: The application definition language was expecting a decimal number, but characters other the 0-9 appeared.

Action: Remove incorrect characters.

-220: BROKER_HLLAPI_FUNCTION_NOT_FOUND

Possible Cause: In the Terminal Launcher configuration, you specified a `HLLAPI.DLL` and the name of the function in the DLL. The name of the function cannot be found in the DLL.

Action: Check you have specified the correct terminal emulator type. Make sure that you entered the HLLAPI function correctly.

-221: BROKER_HLLAPI_OBJECT_UNINITIALISED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-222: BROKER_HLLAPI_DLL_LOAD_FAILED

Possible Cause: Terminal Launcher was unable to load the `HLLAPI.DLL` that you specified. The HLLAPI.DLL for that emulator is looking for other DLL files that do not exist or are not installed for that emulator.

Action: Make sure that the path and file that you are entered for the DLL are correct.

Check the vendor's documentation for information about that emulator.

-223: BROKER_HLLAPI_OBJECT_ALREADY_INITIALISED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-224: BROKER_ERROR_DURING_WINHLLAPICLEANUP

Possible Cause: Terminal Launcher has called the WinHLLAPI cleanup function for a WinHLLAPI emulator.

Action: Check the vendor's documentation for information about that emulator.

-225: BROKER_CANNOT_FIND_WINHLLAPISTARTUP_FUNCTION_IN_DLL

Possible Cause: In the Terminal Launcher configuration, you incorrectly specified that the emulator is a WinHLLAPI emulator.

Action: Make sure that you have specified the correct emulator type.

-226: BROKER_ERROR_DURING_WINHLLAPISTARTUP

Possible Causes: The reason can be the following:

- The terminal emulator does not support the right version of HLLAPI (requires at least V.1.1).
- The attempt to reset a connection to a HLLAPI terminal emulator failed.

Action: Check the vendor's documentation for information about that emulator.

-227: BROKER_CANNOT_FINDWINHLLAPICLEANUP_FUNCTION_IN_DLL

Possible Cause: In the Terminal Launcher configuration, you incorrectly specified that the emulator is a WinHLLAPI emulator.

Action: Make sure you have specified the correct emulator type.

See the Novell Web site for information about configuring specific terminal emulators.

-228: BROKER_BUTTON_NOT_FOUND

Possible Cause: For a Windows single sign-on application, no button exists for the control ID you specified. For example, if you specified Click #3, no button exists for control ID #3.

Action: Specify the correct emulator type.

-230: BROKER_SETPLAT_FAILED

Possible Cause: The regular expression that you supplied in the SetPlat command is invalid.

Action: Check the syntax of the regular expression that you provided.

-231: BROKER_AUTH_CANCEL

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-232: BROKER_UNABLE_TO_START_PROGRAM

Possible Cause: The Run command was unable to find and start the requested program.

Action: Make sure that the executable program exists and that the path is correct.

-234: BROKER_FREE_PLATFORM_SCRIPT_NULL_PTR

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-235: BROKER_VBA_LOGIN_INTERFACE_NOT_IMPLEMENTED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-236: BROKER_CHANGEPASSWORD_INVALID_VARIABLE_SYNTAX

Possible Cause: One of the parameters that you pass to the ChangePassword command must be a variable. The parameter that you provided is not a variable.

Action: Specify a variable.

-237: BROKER_MAD_COMMAND_SET_INVALID_VARIABLE_SYNTAX

Possible Cause: The first parameter that you pass to the Set command must be a variable. The parameter that you provided is not a variable.

Action: Specify a variable.

-239: BROKER_POLICY_SCRIPT_ARG_NUM

Possible Cause: One of the commands in a password policy script has too few or too many arguments.

Action: Include the correct number of arguments.

-240: BROKER_VALID_CHARS_OUTNUMBERED

Possible Cause: A password is unable to satisfy a password policy. This is because the maximum number of allowable characters is less than the minimum number of allowable characters.

Action: Set the maximum number of a particular class of characters to a greater number than the minimum number of specified allowable characters.

-241: BROKER_PASSWORD_LOGIC_ERROR

Possible Cause: You have incorrectly set up a password policy. No password can satisfy all the settings.

Action: Work through each restriction in the password policy, and make sure that one restriction does not contradict another restriction in the policy.

-242: BROKER_EXECEPTION_CHARACHER_FOUND

Possible Cause: You entered a password that contains a character that is not allowed.

Action: Use allowable characters in your password.

-243: BROKER_PASSWORD_TOO_SHORT

Possible Cause: You entered a password that does not have enough characters.

Action: Provide enough characters in your password.

-244: BROKER_PASSWORD_TOO_LONG

Possible Cause: You entered a password that has too many characters.

Action: Enter the correct number of characters.

-245: BROKER_INSUFFICIENT_UPPERCASE_CHARS

Possible Cause: You entered a password that has too few uppercase characters.

Action: Use the specified number of uppercase characters in your password.

-246: BROKER_TOO_MANY_UPPERCASE_CHARS

Possible Cause: You entered a password that has too many uppercase characters.

Action: Use the specified number of uppercase characters in your password.

-247: BROKER_INSUFFICIENT_LOWERCASE_CHARS

Possible Cause: You entered a password that has too few lowercase characters.

Action: Use the specified number of lowercase characters in your password.

-248: BROKER_TOO_MANY_LOWERCASE_CHARS

Possible Cause: You entered a password that has too many lowercase characters.

Action: Use the specified number of lowercase characters in your password.

-249: BROKER_INSUFFICIENT_PUNCTUATION_CHARS

Possible Cause: You entered a password that has too few punctuation characters.

Action: Use the specified number of punctuation characters in your password.

-250: BROKER_TOO_MANY_PUNCTUATION_CHARS

Possible Cause: You entered a password that has too many punctuation characters.

Action: Use the specified number of punctuation characters in your password.

-251: BROKER_INSUFFICIENT_NUMERALS

Possible Cause: You entered a password that has too few numerals.

Action: Use the specified number of numerals in your password.

-252: BROKER_TOO_MANY_NUMERALS

Possible Cause: You entered a password that has too many numerals.

Action: Use the specified number of numerals in your password.

-253: BROKER_NT_FILE_TRAITS_OP_NOT_IMPLEMENTED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-256: BROKER_UNABLE_TO_GET_NT_CHACE_DIR

Possible Cause: You are using Windows NT 4 Domains mode, but you have not defined or mapped a Home drive.

Action: Log in as the user to determine whether the Home drive and Home path variables are set. If the variables are not set, use the Windows NT domain administrative tools to set them.

NOTE: Version 3.6 and above do not support Windows NT.

-257: BROKER_UNABLE_TO_CREATE_NT_CACHE_DIR

Possible Cause: The user object did not have rights to create a directory on the user's local drive.

Action: Grant the user object rights to the directory.

-259: BROKER_MUST_BEGIN_WITH_UPPERCASE

Possible Cause: You entered a password that did not begin with an uppercase character.

Action: Enter an uppercase character at the beginning of the password.

-260: BROKER_NO_DATA_STORES_LOADED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-261: BROKER_ENTRY_SRC_OBJECT_MISMATCH

Possible Cause: You are using a platform other than NDS or eDirectory and have moved an object. The directory object that you are reading entries from is not the directory object that the entries were saved to.

Action: Manually copy and paste the scripts between the objects.

-262: BROKER_CACHE_FILE_INCORRECT_VERSION

Possible Cause: The cache file that you are trying to load was created by a later version of SecureLogin.

Action: Use the version of SecureLogin that created the cache file.

Install the latest version of SecureLogin.

-263: BROKER_DDE_LOGIN_INTERFACE_NOT_IMPLEMENTED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-264: BROKER_DDE_CONNECT_FAILED

Possible Cause: Terminal Launcher could not connect to a specified DDE emulator.

Action: Make sure that the emulator launched correctly and the emulator's DDE support is turned on.

-265: BROKER_DDE_DISCONNECT_FAILED

Possible Cause: Failed attempt to disconnect from a DDE-supporting terminal emulator.

Action: See the vendor's documentation.

-266: BROKER_NT_FILE_STORAGE_SAVE_FAILED

Possible Cause: The user object was unable to save to the equivalent of a cache file in the Home directory using Windows NT 4 Domains.

Action: Grant the user object rights so that the user can write files to the Home directory.

NOTE: Version 3.6 and above do not support Windows NT.

-269: BROKER_NOT_A_PASSWORD_POLICY_COMMMAND

Possible Cause: An invalid command was used in a password policy.

Action: Make sure that the command is spelled correctly.

-271: BROKER_PASSWORD_UNACCEPTABLE

Possible Cause: The password did not meet the requirements as specified in password policies.

Action: Enter the password correctly.

-273: BROKER_MSTELNET_OPERATION_NOT_SUPPORTED

Possible Cause: The generic emulator cannot support a particular operation, for example, SetCursor.

Action: Do not use the command for generic emulators.

-279: BROKER_EMULATOR_LAUNCH_FAILED

Possible Cause: In Terminal Launcher, you can configure the path to the executable that will run. However, the specified executable is unable to run.

Action: Make sure the path to the emulator is correct.

-280: BROKER_UNABLE_TO_CREATE_EMULATOR

Possible Cause: You have specified an invalid terminal type in TLAUNCH.INI (or the Terminal Launcher configuration).

Action: Specify the correct terminal type.

-281: BROKER_INVALID_CHARACTER_FOUND_IN_PASTE_ID_LIST

Possible Cause: A comma does not separate decimal numbers for copy IDs.

Action: For generic emulators, you must specify a set of copy control IDs. Use a comma to separate decimal numbers.

-282: BROKER_INVALID_CHARACTER_FOUND_IN_COPY_ID_LIST

Possible Cause: A comma does not separate decimal numbers for copy IDs.

Action: For generic emulators, you must specify a set of copy control IDs. Use a comma to separate decimal numbers.

-283: BROKER_UNABLE_TO_READ_TLAUCH_INI

Possible Cause: SecureLogin is unable to read the `TLAUNCH.INI` file because the file has been deleted.

Action: Do one of the following:

- Create a blank `TLAUNCH.INI` file.
- Return to the default `TLAUNCH.INI` file by reinstalling SecureLogin.

-284: BROKER_NO_TERMINAL_TYPE_DEFINED

Possible Cause: The `TLAUNCH.INI` file contains an error. The terminal type for the emulator is not defined.

Action: Use Terminal Launcher to specify a terminal type for the emulator.

-285: BROKER_EMULATOR_INFO_NOT_FOUND

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-286: BROKER_RELOAD_NOT_ENABLED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-287: BROKER_TERMINAL-CONNECT-TRY-AGAIN

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-289: BROKER_WRONG_OBJECT_TYPE

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-290: BROKER_FILE_LOAD_FAILED

Possible Cause: You do not have enough rights to convert an earlier `TLAUNCH.INI` file to a later format.

Action: Do one of the following:

- Read an earlier `TLAUNCH.INI` file.
- Create a new `TLAUNCH.INI` file.

NOTE: Ask the administrator to assign you necessary rights.

-292: BROKER_DLL_NOT_INITIALISED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-294: BROKER_SETPLAT_VARIABLE_MUST_BE_RUN_TIME

Possible Cause: The first argument to a `SetPlat` argument can be a variable. The variable used is not a runtime variable.

Action: Make the first argument a runtime variable.

-295: BROKER_ERROR_CONDITIONAL_COMMAND_NOT_HANDLED

Possible Cause: SecureLogin does not handle text in the second part of an If command.

Action: Make sure that the command is the one listed and documented correctly.

-297: BROKER_PARSER_ELSE_STATEMENT_FOUND

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-298: BROKER_RAW_MODE_MUST_BE_SECOND_ARG

Possible Cause: For the Click command, you have placed the –X and –Y arguments before –Raw.

Action: If you use –Raw, place it as the first argument.

-299: BROKER_DISALLOWED_REPEATS_EXIT

Possible Cause: You have tried to use repeated characters in a Password Policy that does not allow them.

Action: Avoid repeated characters.

-300: BROKER_DISALLOWED_SEQUENTIALS_EXIST

Possible Cause: You have tried to use sequential characters in a password, but a Password policy does not allow them.

Action: Avoid sequential characters.

-301: BROKER_DISALLOWED_KEYBOARD_ADJACENTS_EXIST

Possible Cause: You entered a password that has an unacceptable sequence of characters.

Action: Enter an approved sequence of characters.

-303: BROKER_CHARACTER-NOT-IN-REQUIRED-POSITION

Possible Cause: You entered a password that does not have a character in a required position.

Action: Enter the password correctly.

-308: BROKER_BAD_POSITION_ARGUMENT

Possible Cause: While calling a SetCursor command, you tried to move the cursor to an invalid position. For example, out of the terminal session’s boundary.

Action: Specify a valid position.

-309: BROKER_ERROR_CONVERTING_POSITION

Possible Cause: The conversion from –X and –Y coordinates for the SetCursor command has failed.

Action: Specify the –X and –Y coordinates for one offset from the top left-hand corner of the screen.

-310: BROKER_NOT_A_WRITABLE_VARIABLE

Possible Cause: You tried to save a new value to a type of variable that cannot be written to.

Action: Use a runtime or normal variable.

-311: BROKER_RUN_SCRIPT_AGAIN

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-312: BROKER_NO_OU_PERIOD_FOUND

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-314: BROKER_COPY_BACKUP_FAILED

Possible Cause: When SecureLogin begins to update the cache file, SecureLogin first copies the current cache file to a file with the same name, but uses the extension `.GOOD`.

SecureLogin was unable to copy the file. The `.GOOD` file is already open because another process is using it.

Possible Cause: You do not have rights to create files in the directory.

Action: Ask the administrator to assign you rights to the directory.

-315: BROKER_GOTO_LABEL_ALREADY_DEFINED

Possible Cause: You have used a `GoTo` command, but the label that you directed it to has already been used.

Action: Remove the second label command.

-316: BROKER_GOTO_LABEL_NOT_DEFINED

Possible Cause: You have used a `GoTo` command, but the label that you directed it to has not been defined.

Action: Define the label.

-317: BROKER_INCORRECT_DATABASE_VERSION

Possible Cause: The version of SecureLogin that you are using does not handle the version of SecureLogin that is stored in the directory.

Action: Upgrade to the latest version of SecureLogin.

-318: BROKER_DIRECTORY_CRC_DOES_NOT_MATCH

Possible Cause: Whenever SecureLogin stores an entry in Microsoft Active Directory, SecureLogin employs a redundancy check. If the redundancy check does not match when SecureLogin reloads the entry, then the data in Microsoft Active Directory has been corrupted.

Action: Troubleshoot Microsoft Active Directory or Microsoft ADAM.

-319: BROKER_DISALLOWED_DUPLICATE_EXIST

Possible Cause: You entered a password that has unacceptable duplicate characters.

Action: Enter the password correctly.

-320: BROKER_GOTO_LIST_ASSERTION

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-321: BROKER_SUBROUTINE_NOT_DEFINED

Possible Cause: A Call command is calling a subroutine that has not yet been defined.

Action: Define the subroutine.

-322: BROKER_UNABLE_TO_FIND_PASSWORD_FIELD

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-323: BROKER_PASSWORD_FIELD_STYLE_NOT_SET

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-324: BROKER_WEB_ACTION_NOT_SUPPORTED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-325: BROKER_ENTRY_MUST_HAVE_NON_NULL_KEY

Possible Cause: You tried to add an application definition or variable that is a blank string.

Action: Provide a name for the application definition or variable.

-326: BROKER_VARIABLE_REQUIRED

Possible Cause: Some commands, for example, ReadText, require a variable to copy the data that they are returning to. The argument must be a variable.

Action: Change the argument to a variable.

-327: BROKER_OBJECT_NOT_FOUND

Possible Cause: Microsoft Active Directory/ADAM library was unable to allocate memory.

Action: Troubleshoot Microsoft Active Directory or Microsoft ADAM.

-328: BROKER_ADS_MEMORY_FAILURE

Possible Cause: The Microsoft Active Directory/ADAM library was unable to allocate memory.

Action: Close one or more applications and try again.

-329: BROKER_ADS_ERROR_GETTING_ATTRIBUTE

Possible Cause: Although data exists in Microsoft Active Directory/ADAM, SecureLogin is unable to read the data.

Action: Troubleshoot Microsoft Active Directory or Microsoft ADAM.

-330: BROKER_ADS_INSUFFICIENT_RIGHTS_TO_DELETE

Possible Cause: When you removed an application definition, SecureLogin tried to delete part of an attribute from Microsoft Active Directory/ADAM. However, you are unable to delete the attribute because you do not have sufficient rights.

Action: The administrator must assign sufficient directory rights for each user object so that the user can modify SecureLogin attributes.

-331: BROKER_ADS_ERROR_DELETING_VALUE

Possible Cause: Microsoft Active Directory/ADAM was unable to delete a value.

Action: Troubleshoot Microsoft Active Directory or Microsoft ADAM.

-332: BROKER_NO_PASSWORD_FIELD_VARIABLE_IN_SCRIPT

Possible Cause: A Web application definition must have at least one Type command that has “password” as the second argument.

The following lines illustrate a typical application definition:

- `Type $Username`
- `Type $Password Password`

However, the application definition has no `Type` command followed by the Password attribute.

Action: Add a Type command followed by the Password attribute.

-333: BROKER_REGEX_GET_REPLACE_STRING_FAILED

Possible Cause: On the RegSplit command, the string that you are running through the regular expression did not match.

Action: Change the regular expression.

-335: BROKER_REGEX_COMPILE_FAILED

Possible Cause: The syntax of the regular expression was incorrect.

Action: Revise the syntax of the regular expression.

-336: BROKER_DIRECTORY_AUTH_DATA_CORRUPT

Possible Cause: The SecureLogin:SSOAuth data attribute has become corrupt.

Action: Contact Novell Support.

-337: BROKER_DES_KEY_NOT_SET

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-338: BROKER_DES_INVALID_BLOCK_LEN

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-339: BROKER_INVALID_ENCRYPTION_TYPE

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-340: BROKER_UNKNOWN_DATABASE_VERSION

Possible Cause: You are using an earlier version of SecureLogin.

Action: Upgrade to the latest version of SecureLogin.

-341: BROKER_USER_KEY_NOT_SET

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-343: BROKER_PRIMARY_KEY-DECRYPT_FAILED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-344: BROKER_SECONDARY_KEY_DECRYPT_FAILED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-345: BROKER_MERGE_WRONG_ENTRY_TYPE

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-348: BROKER_PASSWORD_RESET_DETECTED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-349: BROKER_UNABLE_TO_FIND_SESSION_FILE

Possible Cause: Terminal Launcher could not find a session file for an emulator.

Action: Configure Terminal Launcher with the correct path to the file for the emulator session.

-352: BROKER_AUTH_DATA_INCORRECT

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-353: BROKER_RECURSIVE_SCRIPT_INCLUDE_DETECTED

Possible Cause: While using the Include command, you included an application definition twice.

Action: Only include an application definition once.

-354: BROKER_NETWORK_PASSWORD_INCORRECT

Possible Cause: You have turned on the option to prompt the user for the network password before the user can access options on the taskbar, and the user entered an incorrect password.

Action: Enter the correct password.

-355: BROKER_USER_ABORTED_LOAD_PROCESS

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-356: BROKER_INVALID_CHARACTER_FOUND_IN_STARTUP_ID_LIST

Possible Cause: For generic emulators, you specify the startup control ID.

A comma must separate a list of numbers. You have used a character other than a comma.

Action: Remove unacceptable characters.

-357: BROKER_ERRO_REG_CACHE_NO_DETAILS

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-358: BROKER_ERROR_REG_CHACE_SAVE_FAILED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-359: BROKER_ERROR_REG_CACHE_SPLIT

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-360: BROKER_PASSWORD_VARIABLE_NOT_ALLOWED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-361: BROKER_NMAS_DLL_NOT_AVAILABLE

Possible Cause: SecureLogin cannot pad the DLL file for NMAS™ for use with the AAVerify command.

Action: To use features for AAVerify, install NMAS.

-362: BROKER_NMAS_LEGACY_RELOGIN_NOT_FOUND

Possible Cause: SecureLogin could not find the NMAS relogin function in the DLL for NMAS.

Action: Install the latest version of NMAS.

-363: BROKER_STANDARD_VARIABLE_REQURED

Possible Cause: A ? variable has been used and this command requires a $ variable.

Action: Provide a $ variable.

-364: BROKER_LDAP_LOGIN_CANCELLED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-365: BROKER_LDAP_INIT_FAILED

Possible Cause: The initialization of the LDAP SSL layer failed.

Action: Contact Novell Support.

-367: BROKER_REG_AUTH_CACHE_MISMATCH

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-368: BROKER_LDAP_TOKEN_DELETED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-369: BROKER_CRED_LIST_NOT_NULL

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-370: BROKER_CRED_LIST_NULL

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-371: BROKER_NO_MORE_CRED_SETS

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-372: BROKER_ACCESS_IS_DENIED

Possible Cause: For LDAP, you do not have rights to the area of the directory that you are trying to access.

Action: Grant user objects the correct rights.

-373: BROKER_HLLAPI-CONNECT_FAILED

Possible Cause: Terminal Launcher was unable to connect to the emulator.

Action: Make sure that the emulator has HLLAPI enabled.

-374: BROKER_DUPLICATE_ENTRIES_EXIST

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-375: BROKER_NOT_RUNNING_NT

Possible Cause: Although you are not running Windows NT, you tried to use a feature that is available only through Windows NT.

Action: Do not use that feature unless you are running Windows NT.

-376: BROKER_WINNT_CACHE_AUTH_REG_FAILED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-377: BROKER_WINNT_CACHE_AUTH_REG_WRONG_USER

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-378: BROKER_INVALID_PIPE_STRING_FOUND

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-379: BROKER_HEX_LENGTH_INCORRECT

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-380: BROKER_HLLAPI_NOT_CONNECTED_TO_PS

Possible Cause: Terminal Launcher tried to use a HLLAPI function. However, the HLLAPI DLL is not connected to the emulator presentation space.

Action: Make sure that Terminal Launcher is set up correctly with the emulator.

-381: BROKER_HLLAPI_SPECIFYING_PARAMETERS_ERROR

Possible Cause: Incorrect parameters were given to a command that uses a HLLAPI function.

Action: Contact Novell Support.

-382: BROKER_HLLAPI_INVALID_PS_POSITION

Possible Cause: An attempt was made to move the cursor or read text from an invalid (out of bounds) position on the emulator presentation space.

Action: Correct the positioning parameter in the application definition.

-383: BROKER_HLLAPI_SYSTEM_ERROR

Possible Cause: Terminal Launcher is not configured correctly for the emulator.

Action: Make sure that Terminal Launcher is set up correctly with the emulator and that the emulator correctly supports HLLAPI.

-384: BROKER_HLLAPI_PS_BUSY_ERROR

Possible Cause: A HLLAPI function is being called while the emulator presentation space is unavailable.

Action: Make sure that the emulator is not being used by another HLLAPI application.

-385: BROKER_HLLAPI_INPUT_REJECTED

Possible Cause: The emulator rejected an attempt to input data into the emulator presentation space.

Action: Make sure that the emulator presentation space is not locked.

-386: BROKER_HLLAPI_ERROR_QUERYING_SESSIONS

Possible Cause: SecureLogin is unable to query available HLLAPI sessions.

Action: Make sure that the Terminal Launcher is set up correctly with the emulator.

-387: BROKER_LAST_NDS_USER_NOT_FOUND

Possible Cause: The last NDS or eDirectory user object, as stored in the registry, could not be read for use in an NMAS logon.

Action: Make sure the last NDS or eDirectory user object is stored correctly in the registry.

-388: BROKER_LAST_NDS_USER_UNWORTHY

Possible Cause: The last NDS or eDirectory user object, as stored in the registry, was not in the correct format. An NMAS logon was unable to use the format.

Action: Make sure the last NDS or eDirectory user object is stored correctly in the registry.

-389: BROKER_NMAS_DISCONNECTED_LOGIN_NOT_FOUND

Possible Cause: NMAS disconnected logon function not found in `NMAS.DLL`.

Action: Make sure that the correct `NMAS.DLL` is installed.

-390: BROKER_LDAP_SSL_INIT_FAILED

Possible Cause: SecureLogin could not initialize the LDAP SSL libraries.

Action: Contact Novell Support.

-391: BROKER_LDAP_SSL_ADD_CERT_FAILED

Possible Cause: SecureLogin could not open the certificate you supplied for LDAP over SSL. Either the file does not exist or it is in the incorrect format. If the certificate file specified ends in `.DER`, then SecureLogin uses Distinguished Encoding Rule (DER) format. Otherwise SecureLogin uses B64 format.

Action: Make sure that the path to the certificate is correct and that it is in DER format.

-392: BROKER_BUILTIN_VARIABLE_NOT_FOUND

Possible Cause: A built-in variable such as ?sysversion was not found.

Action: Check that the variable name is correct.

-393: BROKER_SCRIPT_NOT_PURELY_INDEXED

Possible Cause: While working with Web modules, you mix indexed and nonindexed commands.

For example, you entered the following:

```
Type $Username #1

Type $Password
```

Action: Make sure that all commands use indexes, or remove all indexes.

-394: BROKER_LDAP_PASSWORD_INCORRECT

Possible Cause: The password supplied to log in to LDAP was incorrect.

Action: Check the password.

-395: BROKER_LDAP_USER_NON_EXISTANT

Possible Cause: The user name that you used to log on to LDAP does not exist.

Action: Make sure that the user name exists in the directory and that the LDAP context is correct.

-396: BROKER_LDAP_SERVER_DETAILS_INCORRECT

Possible Cause: One or more of the LDAP server parameters supplied was incorrect.

Action: Check the LDAP server address and port number.

Make sure that the LDAP server you are connected to is running.

-398: BROKER_WIZ_CP_WRONG_SCRIPT_TYPE

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-399: BROKER_DIVIDE_BY_ZERO_IS_BAD

Possible Cause: Using the Divide command, you attempted division by zero.

Action: Do not attempt division by zero.

-400: BROKER_WRONG_SECTION_NAME

Possible Cause: You manually edited a wizard-generated application definition.

Action: When editing an application definition, do not edit the specially generated comments. Only edit the actual commands. If this error occurs, you will no longer be able to use the wizard for that application definition.

-401: BROKER_INVALID_GLOBAL_WIZARD_CONFIG

Possible Cause: You manually edited a wizard-generated application definition.

Action: Do not edit the specially generated comments in an application definition. Only edit the actual commands. If this error occurs, you will no longer be able to use the wizard for that application definition.

-402: BROKER_LDAP_ATTRIBUTE_DOES_NOT_EXIST_IN_SCHEMA

Possible Cause: Either of the following:

- You are running LDAP on eDirectory, but have not correctly mapped the LDAP attributes.
- You are running LDAP on a platform other than eDirectory. However, the schema is not extended for that platform.

Action: Check your LDAP attribute mappings. Extend the LDAP schema.

-403: BROKER_AAVERIFY_DLL_NOT_AVAILABLE

Possible Cause: SecureLogin was unable to load `SL_AAVERIFY.DLL`.

Action: Make sure that you have the correct DLLs installed for `AAVERIFY`.

-404: BROKER_AAVERIFY_FUNCTION_NOT_FOUND

Possible Cause: You are using the incorrect version of `SL_AAVERIFY.DLL`.

Action: Check the version of `SL_AAVERIFY.DLL`.

-405: BROKER_AAVERIFY_CONSISTENCY_FAILURE

Possible Cause: You are using the incorrect version of `SL_AAVERIFY.DLL`.

Action: Check the version of `SL_AAVERIFY.DLL`.

-406: BROKER_AAVERIFY_ERROR

Possible Cause: You are using the incorrect version of `SL_AAVERIFY.DLL`.

Action: Check the version of `SL_AAVERIFY.DLL`.

-408: BROKER_DES_KEY_DATA_CORRUPT

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-409: BROKER_OPERATION_ABORTED_BY_USER

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-410: BROKER_NOT_A_STRING_ATTRIBUTE

Possible Cause: You are using % variables, but the attribute you are reading is not a plain string attribute (SYN_CE_STRING or SYN_CI_STRING on eDirectory).

Action: Check the schema definition of the attribute to confirm that the syntax is SYN_CE_STRING or SYN_CI_STRING.

-411: BROKER_LDAP_INVALID_DN_SYNTAX

Possible Cause: The format of your LDAP user name was invalid.

Action: Check the format of the user name that you entered.

-412: BROKER_INVALID_OPTION_COMBINATION

Possible Cause: An invalid combination of options was passed to an application definition command.

For example, you passed –Right and –Raw to the `Click` command.

Action: See the appropriate application definition command.

-413: BROKER_AAVERIFY_SLOGIN_DOES_NOT_EXIST

Possible Cause: `SL_AAVERIFY.DLL` generates these errors. There is a problem connecting to the SecureLogin server.

Action: Troubleshoot service location problems by reviewing documentation on SecureLogin Advanced Authentication.

-414: BROKER_AAVERIFY_ERR_SLOGIN_NOT_RUNNING

Possible Cause: `SL_AAVERIFY.DLL` generates these errors. There is a problem connecting to the SecureLogin server.

Action: Troubleshoot service location problems by reviewing documentation on SecureLogin Advanced Authentication.

-415: BROKER_AAVERIFY_ERR_LOAD_LIB_SLPAM

Possible Cause: `SL_AAVERIFY.DLL` generates these errors. There is a problem connecting to the SecureLogin server.

Action: Troubleshoot service location problems by reviewing documentation on SecureLogin Advanced Authentication.

-416: BROKER_WI_GETEXENAME_ERR

Possible Cause: The wizard was unable to retrieve the executable name for the window you selected.

Action: Do not use the wizard for this application.

-417: BROKER_ADS_PUT_OCTET_INSUFFICIENT_RIGHTS

Possible Cause: You do not have sufficient rights to Microsoft Active Directory/ADAM to perform the current operation.

Action: Ask the directory administrator to assign you additional Microsoft Active Directory/ADAM system rights.

-418: BROKER_ADS_CLR_OCTET_INSUFFICIENT_RIGHTS

Possible Cause: You do not have sufficient rights to Microsoft Active Directory/ADAM to perform the current operation.

Action: Ask the directory administrator to assign you additional Microsoft Active Directory/ADAM system rights.

-420: BROKER-_SLAASSO_ERR_CRYPTO_KEY_NOT_SET

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-421: BROKER_SLASSO_ERR_UNKNOWN_DATA

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-422: BROKER_SLASSO_OUT_OF_MEMORY

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-423: BROKER_ERROR_INITIALISING_DATA_STORES

Possible Cause: SecureLogin was unable to initialize either the primary or secondary datastore.

Action: Contact Novell Support.

-424: BROKER_UNABLE_TO_LOAD_SLOTP_DLL

Possible Cause: `SLOTP.DLL` could not be loaded. This DLL is required for synchronizing one-time password to LDAP directories.

Action: Review documentation for one-time passwords.

-425: BROKER_LDAP_NO_SUCH_ATTRIBUTE

Possible Cause: You have used a % variable on LDAP. However, the requested attribute does not exist.

Action: Check the spelling of the attribute name against the LDAP schema.

-426: BROKER_SYS_VARIABLE_NOT_AVAILABLE

Possible Cause: A system variable, for example, ?syspassword, was requested but was not available. `SLINA.DLL` or `SLNMAS.DLL` must be correctly installed for these variables to function.

Action: Make sure that either `SLINA.DLL` or `SLNMAS.DLL` is installed.

-427: BROKER_USERNAME_UNSUITABLE_FOR_READING_SLINA_CREDS

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-428: BROKER_NO_EXCEPTION_HANDLER_DEFINED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-429: BROKER_EXCEPTOPN_RAISED

Possible Cause: Data has become corrupted, or the software is not working as intended.

Action: Contact Novell Support.

-430: BROKER_MUST_BE_CALL_OR_GOTO

Possible Cause: When using the `OnException` command, the second parameter must be Call or `GoTo`.

Action:

-442: BROKER_CHAR_UCASE_NOT_IN_REQUIRED_POSITION

Possible Cause: There is not an uppercase character in a position where one is required.

Action: Check the password for compliance with the Password Policy.

-443: BROKER_CHAR_LCASE_NOT_IN_REQURED_POSITION

Possible Cause: Raised by the Password Policy code if there is not a lower case character in a position where one is required.

Action: Check the password for compliance with Password Policy.

-444: BROKER_PUNCTUATION_NOT_IN_REQUIRED_POSITION

Possible Cause: There is not a punctuation character in a position where one is required.

Action: Check password for compliance with Password Policy.

-477: BROKER_UNABLE_TO_GET_REGISTRY_DATA

Possible Cause: The SecureLogin application definition `GetReg` command could not read the required registry information.

Action: Contact Novell Support.

-478: BROKER_ERROR_PARSING_PARAMETER

Possible Cause: The registry entry name passed to the SecureLogin application definition `GetReg` command was incorrect.

Action: Make sure that the name begins {HKCR, HKCC, HKCU, HKLM, or HKU} and corresponds to one of the Windows registry hives. Also, it must contain the path to the desired registry entry within the node.

-481: BROKER_AUTH_QUERY_ON_WRONG_OBJECT_TYPE

Possible Cause: SecureLogin has attempted to load data from a directory object of an incorrect type.

Action: Contact Novell Support.

-482: BROKER_VERSION_NO_ROLL_BACK

Possible Cause: The SecureLogin datastore version cannot be returned to an older datastore version after it has been set to version 6.0.

Action: Contact Novell Support.

-483: BROKER_SECURE_CONNECTION_REQUIRED

Possible Cause: SecureLogin cannot load sensitive data from the server over insecure connections.

Action: Contact Novell Support.

-500: BROKER_ERROR-ACCOUNT-EXPIRED

Possible Cause: SecureLogin was unable to authenticate your Active Directory account because your user account has expired.

Action: Contact your system administrator.

-501: BROKER_ERROR_ACCOUNT_DISABLED

Possible Cause: SecureLogin was unable to authenticate your Active Directory account because your user account has been disabled.

Action: Contact your system administrator.

-502: BROKER_ERROR_ACCOUNT_LOCKED

Possible Cause: SecureLogin was unable to authenticate your Active Directory account because your user account has been locked.

Action: Contact your system administrator.

-503: BROKER_ERROR_PASSWORD_EXPIRED

Possible Cause: SecureLogin was unable to authenticate your Active Directory account because your password has expired.

Action: Change your Active Directory password or contact your system administrator.

-600: BROKER_NONFIR_INVALID_TARGET

Possible Cause: A non-directory datastore is unable to load the local rule that contains the required data for an object. This could be because of the following:

- Insufficient user permissions.
- File failed to download.
- File has been deleted.

Action: Contact your system administrator.

-2147016656: Error opening specified object

Possible Cause: Microsoft Active Directory code error message (value 0x80072031): There is no such object on the server.

Action: You have entered an incorrect object or container definition when assigning user rights. Reenter the correct object or container definition.

Schema Updates

B

This section provides information on the following:

B.1 Schema Attributes

SecureLogin adds six schema attributes to the directory. The attributes are added during installation using the appropriate schema extension tool, depending on your choice of directory for SecureLogin data storage. In Active Directory and LDAP environments, `adsschema.exe` is used. For Novell NDS or eDirectory environments, `ndsschema.exe` is used.

These attributes are required for the encryption and storage of SecureLogin data against directory objects such as user objects and organizational units. The following descriptions include the type of data stored for each attribute and the security rights required to permit the data to be saved for the SecureLogin client.

Before installing SecureLogin, you need to extend the directory schema. For information on extending the schema, see “Extending the eDirectory Schema” in the *Novell SecureLogin 6.1 SP1 Installation Guide.*

If you are upgrading from a SecureLogin version older than 3.5, you need to extend your schemas.

B.2 Active Directory Environments

In Active Directory environments, `adsschema.exe` is used.

B.2.1 Protocom-SSO-Auth-Data

This attribute contains all user-specific authentication data, such as the passphrase.

Attribute Name	Protocom-SSO-Auth-Data
Classes assigned to	User
Syntax	Octet String

Optional Flags	Synchronize
X.500 OID	1.2.840.113556.1.8000.60.2

B.2.2 Protocom-SSO-Entries

This attribute contains the following:

- All the user's login credentials, including passwords.
- Specific preferences and application definitions at the user object.
- Corporate application definitions and preferences at the container and organizational unit objects.

Attribute Name	Protocom-SSO-Entries
Classes assigned to	Container
	Organizational Unit
	User
Syntax	Octet String
Optional Flags	Synchronize
X.500 OID	1.2.840.113556.1.8000.60.1

B.2.3 Protocom-SSO-Entries-Checksum

This attribute stores a checksum so that the single sign-on client can easily determine whether a complete reload of single sign-on adapter information is required.

Attribute Name	Protocom-SSO-Entries Checksum
Classes assigned to	Container
	Organizational Unit
	User
Syntax	Octet String
Optional Flags	Synchronize
X.500 OID	1.2.840.113556.1.8000.60.5

B.2.4 Protocom-SSO-Profile

This attribute stores the address of the organizational unit to be redirected to.

Attribute Name	Protocom-SSO-Profile

Classes assigned to	Container
	Organizational Unit
	User
Syntax	Distinguished Name
Optional Flags	Synchronize
X.500 OID	1.2.840.113556.1.8000.60.7

B.2.5 Protocom-SSO-Security-Prefs

This attribute stores the data required for advanced passphrase policies, including administrator set passphrase questions and passphrase help information and settings.

Attribute Name	Protocom-SSO-Security-Prefs
Classes assigned to	Container
	Organizational Unit
	User
Syntax	Octet String
Optional Flags	Synchronize
X.500 OID	1.2.840.113556.1.8000.60.3

B.2.6 Protocom-SSO-Security-Prefs-Checksum

A checksum used to optimize reading of the Security Preference attribute.

Attribute Name	Protocom-SSO-Security-Prefs-Checksum
Classes assigned to	Container
	Organizational Unit
	User
Syntax	Octet String
Optional Flags	Synchronize
X.500 OID	1.2.840.113556.1.8000.60.6

B.3 LDAP Environments

In LDAP environments, `adsschema.exe` is used.

- Section B.3.1, “Protocom-SSO-Auth-Data,” on page 274
- Section B.3.2, “Protocom-SSO-Entries,” on page 274
- Section B.3.3, “Protocom-SSO-Entries-Checksum,” on page 274

B.3.1 Protocom-SSO-Auth-Data

This attribute contains all user-specific authentication data, such as the passphrase.

Attribute Name	Protocom-SSO-Auth-Data
Classes assigned to	User
OID	2.16.840.1.113719.2.26.4.1.1

B.3.2 Protocom-SSO-Entries

This attribute contains the following:

- All the user's logon credentials, including passwords.
- Specific preferences and application definitions at the user object.
- Corporate application definitions and preferences at the container and organizational unit objects.

Attribute Name	Protocom-SSO-Entries
Classes assigned to	Container
	Organizational Unit
	User
OID	2.16.840.1.113719.2.26.4.2.1

B.3.3 Protocom-SSO-Entries-Checksum

This attribute stores a checksum so that the single sign-on client can easily determine whether a complete reload of single sign-on adapter information is required.

Attribute Name	Protocom-SSO-Entries Checksum
Classes assigned to	Container
	Organizational Unit
	User
OID	2.16.840.1.113719.2.26.4.5.1

B.3.4 Protocom-SSO-Profile

This attribute stores the address of the organizational unit to be redirected to.

Attribute Name	Protocom-SSO-Profile
Classes assigned to	Container
	Organizational Unit
	User
OID	2.16.840.1.113719.2.26.4.17.1

B.3.5 Protocom-SSO-Security-Prefs

This attribute stores the data required for advanced passphrase policies including administrator set passphrase questions and passphrase help information and settings.

Attribute Name	Protocom-SSO-Security-Prefs
Classes assigned to	Container
	Organizational Unit
	User
OID	2.16.840.1.113719.2.26.4.4.1

B.3.6 Protocom-SSO-Security-Prefs-Checksum

A checksum used to optimize reading of the Security Preference attribute.

Attribute Name	Protocom-SSO-Security-Prefs-Checksum
Classes assigned to	Container
	Organizational Unit
	User
OID	2.16.840.1.113719.2.26.4.6.1

B.4 Security Rights Assignments

This section contains information on the following:

B.4.1 User-Based Attributes

The directory user objects for people using the SecureLogin requires the following attribute rights against their own objects.

Attribute Name	Entry Rights Required
Protocom-SSO-Auth-Data	Read/Write
Protocom-SSO-Entries	Read/Write
Protocom-SSO-Entries-Checksum	Read/Write
Protocom-SSO-Profile	Read/Write
Protocom-SSO-Security-Prefs	Read/Write
Protocom-SSO-Security-Prefs-Checksum	Read/Write

B.4.2 Container-Based Attributes

In addition, users require the following directory attribute rights against all container objects.

Attribute Name	Entry Rights Required
Protocom-SSO-Entries	Read
Protocom-SSO-Entries-Checksum	Read
Protocom-SSO-Profile	Read
Protocom-SSO-Security-Prefs	Read
Protocom-SSO-Security-Prefs-Checksum	Read